

Trabajo de Fin de Grado

**Grado en Ingeniería en Tecnologías Industriales**

**Optimización de un Proceso de Carga de Inyectables  
Farmacéuticos mediante Brazo Robótico: Diseño,  
Programación del PLC e Implementación**

**MEMORIA**

**Autor:** Gabriel Marino Sanjurjo  
**Director:** Vicenç Puig Cayuela  
**Convocatoria:** Junio 2019



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





## Resumen

De la mano de la rápida evolución que está viviendo la tecnología en la actualidad, la productividad en la industria está creciendo gracias a nuevos métodos tecnológicos, en especial en el campo de la automatización. En el presente proyecto se identifica una situación problemática en una industria y se confecciona una solución a partir de la automatización de dicho proceso.

Concretamente, el proyecto se basa en la mejora de un proceso de carga de recipientes de suero de la empresa farmacéutica Fresenius Kabi, con planta en Vilassar de Dalt (Barcelona). Este proceso se realiza mediante un brazo robótico que transporta el producto desde la salida de su envasado hasta unas bandejas metálicas, en las que se llevarán las botellas a una autoclave para su esterilización. El problema reside en la posición de los envases en el momento en que el robot los coge, ya que no cuentan con la misma separación que tendrán en las bandejas, por lo que el propio robot ha de separarlos para adaptarlos. En ocasiones, esta separación es incorrecta y se produce un fallo en la colocación, provocando posteriores paradas en la línea de condicionamiento.

El objetivo principal es implementar un sistema automático que sea capaz de separar las botellas a medida que van saliendo del envasado para evitar que tenga que realizar esta acción el robot, facilitando la colocación. Para llegar a este automatismo, es imprescindible el uso de un Controlador Lógico Programable (PLC). Así bien, el tema central de este proyecto es la programación del PLC que gobernará el proceso, usándose un autómata de la marca Siemens. Todo ello se complementa con la incorporación de una Interfaz Hombre – Máquina (HMI) gracias a la cual, los operarios son capaces de controlar el proceso.

Para llegar a este programa se emplea la teoría de la Guía GEMMA, una metodología pautada para el diseño de automatismos, junto con el uso de los GRAFCET, diagramas de estados comúnmente utilizados para procesos cíclicos con estados marcados. Por otro lado, también se trabajan ámbitos como la electrónica en cuanto al uso de sensores para la obtención de las variables para el automatismo, la neumática para los actuadores encargados de realizar las separaciones, además del diseño de la estructura y de los planos, como de estudios de viabilidad económica.

El alcance de este trabajo no se limita únicamente al diseño de esta optimización sino que también a su implementación, con la que finalmente se busca obtener los resultados planteados.

# Sumario

<b>Resumen .....</b>	<b>3</b>
<b>Sumario .....</b>	<b>4</b>
<b>1. Glosario .....</b>	<b>6</b>
<b>2. Prefacio .....</b>	<b>8</b>
2.1. Origen del proyecto .....	8
2.2. Motivación .....	8
2.3. Requerimientos previos .....	8
<b>3. Introducción .....</b>	<b>9</b>
3.1. Objetivos del proyecto .....	9
3.2. Alcance del proyecto .....	9
<b>4. Proceso productivo .....</b>	<b>10</b>
4.1. Presentación del proceso .....	10
4.2. Proceso de carga y descarga de viales .....	11
4.3. Toma y análisis de datos .....	13
<b>5. Solución planteada .....</b>	<b>14</b>
5.1. Viabilidad técnica .....	14
5.1.1. Presupuesto inicial .....	14
5.2. Viabilidad económica .....	16
<b>6. Diseño del diagrama de control .....</b>	<b>17</b>
6.1. Definición GRAFCET .....	17
6.2. Guía GEMMA .....	19
6.2.1. Representación de la Guía GEMMA .....	20
6.2.2. Módulo de producción .....	21
6.2.3. Módulo de modos de marcha .....	22
6.2.4. Módulo de seguridad .....	24
<b>7. Controlador Lógico Programable .....</b>	<b>27</b>
7.1. PLC Siemens .....	28
7.1.1. Lenguajes programación para SIMATIC STEP7 .....	29
7.1.2. Instrucciones básicas .....	30
7.1.3. Módulos y áreas de memoria .....	37
7.2. Programa del PLC .....	38
7.2.1. Ocupación de variables .....	38
7.2.2. Creación de los módulos .....	40

<b>8. Supervisión, Control y Adquisición de Datos.....</b>	<b>41</b>
8.1. SCADA en Siemens.....	41
8.2. Interfaz Hombre – Máquina.....	42
<b>9. Diseño físico .....</b>	<b>46</b>
9.1. Electroválvulas .....	46
9.2. Cilindros compactos.....	48
9.2.1. Detectores magnéticos.....	49
9.3. Fococélulas.....	50
9.4. Estructura.....	53
<b>10. Implementación.....</b>	<b>56</b>
10.1. Instalación eléctrica.....	56
10.2. Instalación neumática .....	58
10.3. Instalación mecánica .....	58
10.4. Carga del programa .....	60
<b>11. Puesta en marcha .....</b>	<b>61</b>
11.1 Adaptación con el ciclo previo .....	61
11.2. Parada de la cinta de entrada de viales.....	62
11.3. Comprobación alarmas.....	63
11.4. Actuación ante fallos de fotocélulas de posición.....	63
<b>12. Coste final del proyecto .....</b>	<b>65</b>
<b>13. Impacto medioambiental .....</b>	<b>65</b>
<b>14. Resultados obtenidos.....</b>	<b>66</b>
<b>15. Propuestas de mejora.....</b>	<b>67</b>
<b>Conclusiones .....</b>	<b>69</b>
<b>Agradecimientos.....</b>	<b>70</b>
<b>Bibliografía.....</b>	<b>71</b>
<b>Anexos.....</b>	<b>73</b>

# 1. Glosario

Término	Descripción
ACU 1/2	Áreas de memoria de 4 bytes (dobles palabras) cada uno, del programa STEP7. Se usan fundamentalmente para realizar cálculos matemáticos y para transferir información de un área de memoria a otra.
ADEPA	<i>Agence nationale pour le DÉveloppement de la Productique Appliquée à l'industrie</i> (Agencia nacional francesa para el Desarrollo de la Productividad Aplicada a la Industria).
AWL	<i>Anweisungsliste</i> (Lista de Declaraciones): Lenguaje de texto del programa STEP7 en el que se citan por líneas las diferentes operaciones.
BCD	<i>Binary-Coded Decimal</i> (Decimal Codificado en Binario): Estándar para representar números decimales en sistema binario en el que cada dígito decimal se codifica con una secuencia de cuatro bits.
BP6/8	<i>Bottle-Pack</i> (Envasadora de botellas) 6/8: Nomenclatura interna de la empresa para las extrusoras que envasan los inyectables de suero.
CPU	<i>Central Processing Unit</i> (Unidad Central de Procesamiento): Hardware de un ordenador o dispositivo programable que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entradas/salidas del sistema.
DB	<i>Data Block</i> (Bloque de datos): Áreas de memoria del programa STEP7 que actualiza el usuario pero permanecen durante los ciclos del programa.
DP	<i>Decentralized Periphery</i> (Periferia Descentralizada): Perfil de conexión Profibus optimizado para ofrecer mayor velocidad, eficiencia y bajo costo de conexiones. Diseñado especialmente para comunicación entre sistemas automáticos de control y E/S (entradas/salidas) distribuidos a nivel de campo (periferia distribuida).
FB	<i>Function Block</i> (Bloque de Funciones): Módulo del programa STEP7 en el que se incorporan los distintos segmentos del código. Tiene asociada una DB de instancia con tal de poder almacenar datos.
FC	<i>Function</i> (Función): Módulo del programa STEP7 en el que se incorporan los distintos segmentos del código. Suele usarse para subrutinas que no requieran un almacenaje de datos.
FUP	<i>Funktionsplan</i> (Diagrama de Funciones): Lenguaje gráfico del programa STEP7 que describe las funciones mediante bloques del álgebra booleana.
GEMMA	<i>Guide d'Étude des Modes de Marche et d'Arrêt</i> (Guía de Estudio de los Modos de Marcha y Paro): Metodología desarrollada por la ADEPA para prever todos los estados de un automatismo en la que se contemplan los estados de producción, parada y defecto.
GRAFCET	<i>Graphe Fonctionnel de Commande des Étapes et Transitions</i> (Diagrama de Control con Etapas y Transiciones): Modelo de representación gráfica de los sucesivos comportamientos de un sistema lógico (estados), predefinido por sus entradas y salidas.

HMI	<i>Human – Machine Interface</i> (Interfaz Hombre – Máquina): Panel de control diseñado para conseguir una comunicación interactiva entre operador y proceso/máquina, con la función de transmitir órdenes, visualizar gráficamente los resultados y obtener una situación del proceso/máquina en tiempo real.
Inyectable	Envase contenedor de una solución farmacológica destinada a la administración parenteral (no atraviesa la vía digestiva). Producto fabricado en la planta referente al presente proyecto.
KOP	<i>Kontaktplan</i> (Diagrama de Contactos): Lenguaje gráfico del programa STEP7 basado en los esquemas eléctricos de control clásicos. Es el referente al lenguaje Ladder.
MPI	<i>Message Passing Interface</i> (Interfaz de Paso de Mensajes): Conecta la programación de estaciones (PC), consolas de operador y otros dispositivos de un sistema de control.
OB	<i>Organization Block</i> (Bloque de Organización): Interface entre el sistema operativo de la CPU de un PLC y el programa proporcionado por el usuario.
Pascal	Lenguaje de programación estructurado basado en texto. Su código está dividido en funciones y cuenta con operadores con funciones predefinidas.
PEEK	<i>Polyether ether ketone</i> (Polieteretercetona): Polímero técnico termoplástico semicristalino con resistencia a altas temperaturas y a la abrasión.
PN	Perfil de comunicación para Profinet.
Profibus	<i>PROcess Field BUS</i> (Bus de Campo de Procesos): Permite la comunicación entre dispositivos dentro de una planta industrial con tal de facilitar la transmisión de datos para su control.
Profinet	<i>PROcess Field NET</i> (Red de Campo de Procesos): Estándar técnico de la industria para la comunicación de datos a través de Industrial Ethernet.
PLC	<i>Programmable Logic Controller</i> (Controlador Lógico Programable): Computadora utilizada en la ingeniería automática para automatizar procesos electromecánicos.
PVC	<i>Polyvinyl chloride</i> (Policloruro de vinilo): Polímero que puede encontrarse en forma rígida o flexible con una alta resistencia a la abrasión.
RLO	<i>Result of Logical Operation</i> (Resultado de Operación Lógica): Variable que almacena el valor de una operación binaria.
SCADA	<i>Supervisory Control And Data Acquisition</i> (Supervisión, Control y Adquisición de Datos): Software para controlar y supervisar procesos industriales a distancia.
SCL	<i>Structured Control Language</i> (Lenguaje Estructurado de Control): Lenguaje de texto del programa STEP7 basado en el lenguaje de alto nivel Pascal.
STEP7	Software de programación de PLC de la marca Siemens.
TIA Portal	<i>Totally Integrated Automation</i> (Automatización Totalmente Integrada): Software de la marca Siemens para la programación de PLC que integra el nivel de manejo, operador, controlador y campo.
Vial	Envase contenedor de una solución farmacológica destinada a la administración parenteral (no atraviesa la vía digestiva). Producto fabricado en la planta referente al presente proyecto.
WinCC	<i>Windows Control Center</i> (Centro de Control de Ventanas): Software SCADA de la marca Siemens.

## 2. Prefacio

### 2.1. Origen del proyecto

Este proyecto surge a partir de unas prácticas en la empresa farmacéutica Fresenius Kabi de Vilassar de Dalt (Barcelona). En ellas, el objetivo principal es realizar una mejora en alguna de las estaciones de la planta de producción, adquiriendo nuevos conocimientos técnicos o ampliando aquellos aprendidos durante el grado y, además, desarrollar aptitudes que permitan trabajar en proyectos del mundo laboral tales como el trabajo en equipo, la planificación, la investigación, etc.

El proyecto se ha centrado en la mejora del proceso de carga de botellas (llamadas viales o inyectables a lo largo del documento) de suero mediante un brazo robótico, a partir de la incorporación de un mecanismo que favorezca dicha acción. Para ello ha sido necesario trabajar en el ámbito mecánico en el diseño tanto de la estructura como de elementos que forman el dispositivo y, en especial, formarse en el campo de la programación del Controlador Lógico Programable (PLC) que gobierna el sistema.

### 2.2. Motivación

Los PLC son elementos muy utilizados en la industria, por lo que es muy importante para un ingeniero industrial saber manejarlos y conocer sus métodos de programación.

Por otro lado, la oportunidad de trabajar en un proyecto dentro de una empresa resulta una experiencia muy enriquecedora para un estudiante. Esto conlleva seguir unas pautas internas, cumplir con plazos y presupuestos, lidiar con imprevistos, tratar con proveedores y compañeros, entre muchas otras situaciones habituales en el mundo laboral. Otro punto importante era el de poder contar con los recursos de la empresa.

Ante tal coyuntura, se quiso aprovechar para realizar el Trabajo de Fin de Grado dentro de la misma empresa en la que se realizaron las prácticas.

### 2.3. Requerimientos previos

A lo largo del Grado en Ingeniería en Tecnologías Industriales impartido por la *Escola Tècnica Superior d'Enginyeria Industrial de Barcelona* (ETSEIB) de la *Universitat Politècnica de Catalunya* (UPC) no se trabaja la programación de PLC más allá de nociones básicas sobre autómatas programables en la asignatura de *Electrónica* o sobre controladores en la materia *Control Automático*, por lo que era necesaria una formación por parte del estudiante. Aun así, los conocimientos de programación en *Python* adquiridos durante el Grado resultan útiles a la hora de conocer las reglas básicas de la programación, por más que se trate de otro lenguaje.

Además, el trabajo también cuenta con una parte de diseño mecánico, en la que pueden ser útiles asignaturas cursadas como *Expresión Gráfica* a la hora de hacer el diseño en 3D y los planos y *Teoría de Máquinas y Mecanismos* en cuanto a las estructuras. Por último, resulta práctico el temario aprendido en *Organización y Gestión* y en *Gestión de Proyectos*, además de la propia experiencia del estudiante en otras prácticas realizadas, de cara a la metodología de trabajo dentro de una empresa.



### 3. Introducción

Fresenius Kabi es uno de los cuatro segmentos de negocios del grupo farmacéutico alemán Fresenius, especializado en medicamentos vitales y tecnologías para la infusión, transfusión y nutrición clínica. La empresa cuenta con más de 36.000 trabajadores repartidos en múltiples países alrededor del mundo.

La planta industrial de Vilassar de Dalt está dedicada a la fabricación de productos de nutrición clínica por vía parenteral (intravenosa) y enteral (gastrointestinal) como por ejemplo el suero fisiológico, es decir, una solución de Cloruro de Sodio (NaCl) al 0,9%. En la planta se produce la disolución (a partir de agua y el soluto correspondiente) además de las botellas (a partir de granza de polietileno), para posteriormente empaquetarse. Uno de los formatos más producidos en la planta es el de inyectables de 10 ml.

Dado que la planta produce 24 horas los siete días de la semana y este es un producto con una alta demanda, cualquier parada de la línea de producción tiene una gran repercusión en las ventas de la empresa, causándole pérdidas. Es por esto que, incluso teniendo una baja tasa de paradas o de rechazo de producto, se llevan a cabo procesos de mejora constantemente.

#### 3.1. Objetivos del proyecto

El objetivo básico de este proyecto es identificar algún elemento de la línea de producción y condicionado de viales de 10 ml que sea propicio a ser optimizado y diseñar e implementar dicha mejora de manera económica y lo más inmediata posible ya que la duración de las prácticas es de cinco meses.

Con esta solución se espera recuperar la inversión realizada gracias a un aumento de la producción y además, poder formar en el campo de la programación y de la mecánica, entre otros, al trabajador (en este caso, el estudiante).

#### 3.2. Alcance del proyecto

En el actual trabajo se persigue tratar de manera integradora diversos ámbitos que pueden aparecer en un proyecto industrial, comenzando por una toma y análisis de datos para determinar qué problemas están sucediendo en la línea de producción y cuáles son de mayor interés o de actuación más directa.

A partir de este análisis, se quiere encontrar una solución viable tanto económica como técnicamente, por lo que se requiere un breve análisis económico de la propuesta. Una vez validada, se efectuará el diseño mecánico y eléctrico de la estructura a implementar y, además, la programación del PLC que gobernará dicha optimización será de diseño propio.

Finalmente, será importante el trato con proveedores, compañeros de un mismo o distinto departamento y la puesta en marcha de la modificación, junto con la verificación de los resultados obtenidos y la documentación posterior asociada a cualquier reforma.

## 4. Proceso productivo

### 4.1. Presentación del proceso

Tal y como se ha comentado anteriormente, en la misma planta se producen los envases y las soluciones que estos contendrán. En la primera etapa del proceso se realizan los inyectables mediante una extrusión con soplado. En la *Figura 4.1* puede apreciarse el aspecto del formato de ampollas de 10 ml tras salir del envasado.



**Figura 4.1:** Tira de inyectables de 10 ml de producto a la salida de envasado

Por normativa, es necesario esterilizar el producto. Este proceso se lleva a cabo en una autoclave en la que se regulará la temperatura y presión. Para entrar los inyectables a la autoclave, se posicionan en unas bandejas metálicas que se van apilando hasta formar un bloque como el mostrado en la *Figura 4.2*.



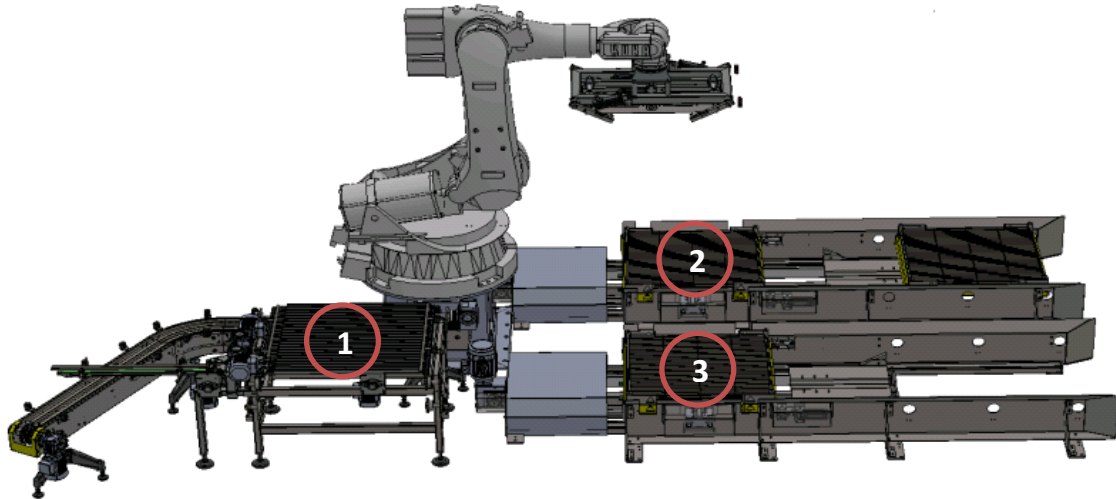
**Figura 4.2:** Bloque de bandejas para la autoclave

En la autoclave, la disolución permanece durante 73 minutos a una temperatura de 112,1°C con una presión constante con tal de eliminar todas las formas de vida microbianas.

Tras salir del esterilizado, se introducen los viales en la línea de condicionado. Ahí se comprueba que el nivel de partículas en suspensión del producto sea inferior al límite establecido, y que los viales sean herméticos para evitar pérdidas. Finalmente, se etiquetan, se introducen en los estuches en los que serán comercializados y se paletizan.

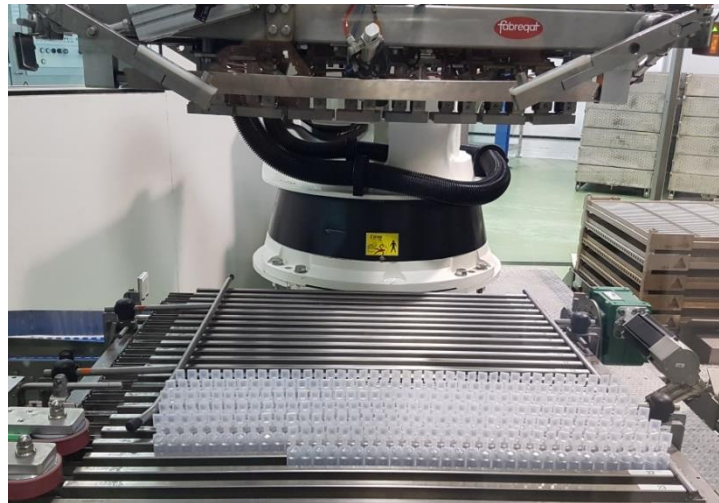
## 4.2. Proceso de carga y descarga de viales

Al inicio de las prácticas se determinó que la estación que iba a estudiarse sería la del proceso de carga de inyectables después de salir del envasado, cuya instalación se esquematiza en la *Figura 4.3*, ya que en ella se utiliza un brazo robótico para ir posicionando los viales en las bandejas y también para ir apilando estas bandejas hasta formar un bloque. El hecho de contar con un brazo robótico resultó de alto interés para el estudio.



**Figura 4.3:** Estación de carga de viales

Los viales salen por la parte izquierda instalación (*Figura 4.3*) y se van posicionando en filas de ocho tiras de inyectables (grupos de 5 botellas) en la mesa de acumulación representada con el número 1 (*Figura 4.3*), tal y como puede verse en la *Figura 4.4*.

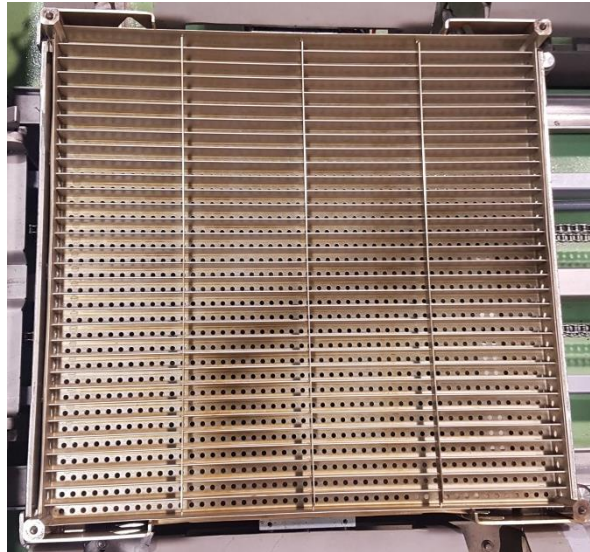


**Figura 4.4:** Mesa de acumulación de viales para su carga

A medida que se llena cada una de las filas, los compartimentos de la mesa van avanzando y el brazo robótico coge filas de dos en dos para ir posicionándolas en la bandeja superior ubicada en la posición 3 de la *Figura 4.3*. Cuando una bandeja se llena, el brazo coge la bandeja superior de la posición 2 de la *Figura 4.3* y la coloca de nuevo en la posición 3. Este proceso se repite sucesivamente hasta que se pasa de un bloque sin viales en la posición 2 a uno lleno en la posición 3.



El problema surge en la configuración de las bandejas. Debido a la necesidad de dar rigidez a las filas de estas, se requieren tres refuerzos verticales como los que se muestran en la *Figura 4.5*, dividiendo las filas en cuatro compartimentos que contarán con dos tiras de inyectables cada uno.



**Figura 4.5:** Bandeja para posicionar los viales al entrar en la autoclave

Visto que los viales salen del envasado de forma continua y quedan juntos en la fila, el robot, tras coger mediante pinzas dos filas de botellas de la mesa, ha de separar dichas pinzas en cuatro grupos de dos tiras para adaptarse a la forma de la bandeja. A pesar de que el sistema es efectivo en la mayoría de los casos, hay situaciones en las que el robot no coge correctamente los viales y estos no entran en la bandeja, quedando torcidos como se demuestra en la *Figura 4.6*.



**Figura 4.6:** Ejemplo de vial mal posicionado

Este suceso no afecta al transcurso de la carga ya que al colocar la siguiente bandeja, el propio peso de esta aplasta el vial mal colocado, pero sí que afecta en la descarga. Cuando ya se ha esterilizado el producto, otro robot coge los inyectables para introducirlos a la línea de condicionado. El aplastamiento deforma el cuello de las botellas, que es el punto por donde las coge el brazo robótico y a causa de esto, el brazo es incapaz de cogerlas, produciendo una parada en la línea hasta que un operario entra a la estación de descarga y retira el inyectable defectuoso.

### 4.3. Toma y análisis de datos

Una vez determinado cuál sería el punto de la línea de producción a estudiar, el siguiente paso era analizar la situación en la que se encontraba el proceso y determinar qué tasa de fallos tenía.

La estación descrita en el *Apartado 4.2* se encuentra por partida doble en la planta, ya que hay dos extrusoras que realizan el producto (BP6 y BP8). Para tomar los datos se dispuso de dos semanas (30 horas semanales) en las que se observarían simultáneamente ambas máquinas, anotando cada vez que ocurriese un fallo como el citado anteriormente, adjuntando además en qué fila, posición y bandeja ocurría.

De los datos recopilados a lo largo de esas dos semanas se pudo determinar que la tasa de este tipo de fallos. Estos valores se representan en promedio e indican el porcentaje de inyectables colocados torcidos en función del total de unidades colocadas a lo largo del estudio.

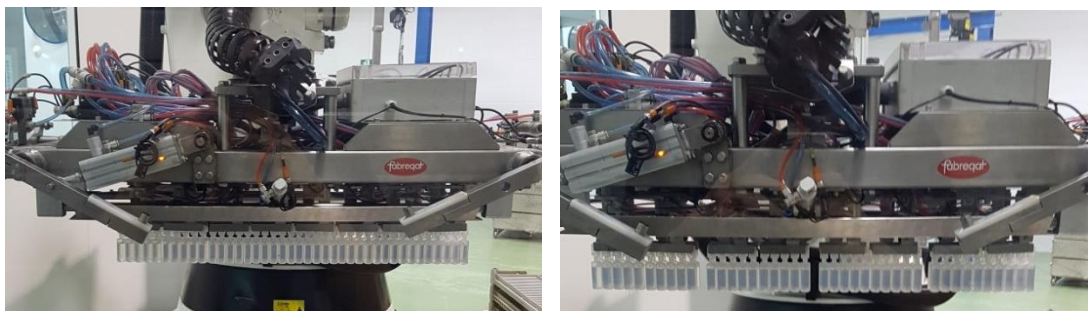
**Tabla 4.1:** *Porcentaje de viales torcidos*

	BP6	BP8
Porcentaje de viales mal colocados	0,2126	0,1291

Por otro lado, una segunda conclusión a la que se pudo llegar fue que la mayoría de las veces que los viales eran mal posicionados, ocurría en un extremo de la bandeja, siendo el mismo para ambas máquinas.

La explicación que se le dio a este suceso fue que los viales no siempre quedan igual colocados en las filas de la mesa. Esto es porque en una misma tira, las cinco botellas no quedan nunca completamente alineadas ya que la unión no es completamente rígida y sufren cierta flexión, por lo que la longitud de las tiras no es siempre la misma en la mesa. Esta variabilidad se ve acentuada al acumularse ocho tiras en cada fila y afecta a la separación que efectúa el robot para colocarlas en las bandejas.

Para conseguir esta separación, no todas las pinzas tienen el mismo desplazamiento, ya que las dos de la derecha (según la *Figura 4.7a*) permanecen quietas mientras que las colindantes se desplazan una distancia mayor cuanto más a la izquierda están. Es en la pinza de la izquierda en la que se observan más fallos, ya que es la que realiza un mayor desplazamiento. La variación puede ser de pocos milímetros pero al tratarse de dimensiones tan reducidas, acaba siendo determinante.



**Figuras 4.7a y 4.7b:** *Pinzas de carga juntas/separadas*

## 5. Solución planteada

Al descubrir que el problema podía venir de la separación de las tiras de inyectables, se quiso idear una solución que permitiese evitar que la tuviese que realizar el brazo robótico. Para ello, se tendría que producir en la misma mesa de acumulación.

Tras plantear distintas alternativas, se creyó conveniente diseñar un mecanismo que fuese capaz de ir separando en grupos de dos tiras de viales a medida que se fuesen entrando a la mesa. Para ello, era necesario incorporar unos topes que sirviesen como separación entre los grupos de dos, asemejándose a la separación que provocan los refuerzos de las bandejas.

La dificultad radicaba en que este sistema de topes debía ser capaz de salir en el momento que hubiese pasado un número par de tiras y entrar en el momento en que avanzase una fila de la mesa para permitir el paso de los viales de la siguiente.

Para ello, era imprescindible el uso de un Controlador Lógico Programable (PLC), que ya estaba presente en la instalación pero que requeriría un nuevo código.

### 5.1. Viabilidad técnica

Para conseguir los topes mencionados anteriormente podían usarse tres cilindros neumáticos (uno para cada separación) que se colocarían en frente de la mesa y que a medida que la primera fila se fuese llenando de viales, irían saliendo por orden para retener los inyectables en la posición adecuada, que sería con la misma separación que en la bandeja, con tal de que el robot únicamente tuviese que coger los viales y colocarlos del mismo modo.

Tras haber salido las ocho tiras de inyectables que conforman cada fila, la mesa avanzaría del mismo modo que hacía en el previo proceso, por lo que los pistones deberían entrar para dejar paso a las botellas de la nueva fila. Con tal de no perder la separación, habría que incorporar unas guías en la parte posterior de la mesa para conseguir tener cuatro carriles, que serían iguales a los cuatro compartimentos que tienen las bandejas.

Para asegurar el correcto funcionamiento del dispositivo, se incorporaría una fotocélula al lado de cada uno de los cilindros para comprobar la llegada de los viales. Los cilindros serían biestables y deberían contar con sensores magnéticos de fin de carrera en ambas posiciones.

#### 5.1.1. Presupuesto inicial

Con estas condiciones, se contactó con una empresa de instalaciones y servicios para proveer el material y proporcionar asistencia en la instalación eléctrica, neumática y de la estructura. Esta última sería de perfil cuadrado ranurado de aluminio para poder ajustarse de manera rápida.

Respecto a la programación, dependería únicamente de la propia empresa (en este caso, del estudiante).

A partir de estas especificaciones, se llegó a un primer presupuesto (precio individual por máquina).

### **PRESUPUESTO P-190226-1FRESEN**

Premià de Dalt, 26 de Febrero del 2.019

Muy Sres. nuestros:

A continuación les pasamos nuestra mejor oferta para la instalación y montaje de 3 cilindros para topes de posición de tiras de inyectables en carga BP6.

Este presupuesto incluye los puntos siguientes:

- Suministro y montaje de una estructura en perfil ranurado de aluminio de 35x35mm.
- Mecanizado de 3 pletinas para sujeción de cilindros.
- Mecanizado de 3 pletinas para sujeción de fotocélulas.
- Mecanizado de topes con material plástico (PEEK).
- Montaje en máquina con sujeción a bancada y suelo del soporte de aluminio.
- Suministro de 3 fotocélulas de IFM OJ5158.
- Suministro y montaje de una batería de 3 electroválvulas biestables de SMC VQC2000/ F-Kit.
- Suministro de 3 cilindros neumáticos CDQS B25-45DC.
- Suministro de 6 detectores magnéticos D-M9PSAPC-595.
- Conexión al armario de las 6 salidas.
- Alimentar neumáticamente el bloque.
- Entrega esquemas reflejando la conexión de estas válvulas.

Total Presupuesto.....3.527 €

El IVA no está incluido en este presupuesto

Validez de la oferta: 3 meses

Condiciones de pago: 35% a la confirmación del pedido  
65% al finalizar el trabajo

## 5.2. Viabilidad económica

Puesto que el impacto de la pérdida por aplastamiento de una tira de inyectables era mínimo en comparación con las pérdidas por tener la línea parada, se estudió qué cantidad de tiempo representaba cada parada a causa de un vial deformado por haberse colocado torcido. En la planta se cuenta con dos extrusoras que fabrican el producto pero únicamente hay una línea de condicionado, por lo que un fallo como el mencionado provocaba una parada total de la producción de inyectables de 10 ml.

Al tratarse de un proyecto nuevo, se quiso realizar la prueba únicamente con una de las dos máquinas y, en caso de funcionar, aplicar la misma mejora a la segunda máquina. La máquina con la que se trabajaría se denomina BP6 (Bottle-Pack 6), siguiendo un sistema de nomenclatura interno de la empresa. Hay que considerar que se condicionan durante doce horas al día los inyectables producidos en la BP6 (las otras doce horas se dedican para el mismo producto pero de la BP8, es decir, la otra inyectora).

Se midió el tiempo que tardaba un operario entre que saltaba la alarma, se desplazaba hacia la instalación, retiraba el producto dañado (en muchas ocasiones había provocado que el robot tampoco pudiese coger correctamente los viales cercanos y se volcasen sobre el resto) y rearmaba la máquina para reanudar la producción. De media, el tiempo de parada resultó ser de 30 segundos.

A partir de estos datos se llega a la valoración económica mostrada en la *Tabla 5.1*. Todos los cálculos se realizan considerando como unidades las tiras de inyectables, es decir, un grupo de cinco botellas.

**Tabla 5.1:** Valoración económica del proyecto

	Unidades	Valor	
Producción BP6	Tiras/Hora	3072	
Defectos posicionado	Tiras/Hora	6,531	* 0,002126 Def/Prod
Tiempo paro/producción	Minuto/Hora	3,266	* 0,5 Minuto/Defecto
	Hora/Día	0,6531	* 12 Hora/Día
	Hora/Año	212,26	* 325 Día/Año
Pérdidas económicas	Euro/Año	4245,20	* 20 Euro/Hora
Periodo retorno	Año	0,831	

Una de las exigencias era que el periodo de retorno fuese inferior a los doce meses. Con los primeros cálculos se obtenía un valor de aproximadamente diez meses, por lo que esta premisa se podía cumplir, permitiendo además un margen por si surgía algún sobrecoste durante el proyecto.

Por otro lado, el gasto no representaba una cantidad excesivamente elevada, ya que la empresa cuenta con unos fondos anuales para cada departamento destinados a las inversiones en mejoras.

Así bien, se decidió aceptar el presupuesto. Con ello se iniciaba la toma de medidas para pasar de un primer presupuesto general a una lista más detallada del material requerido.



## 6. Diseño del diagrama de control

Una vez aprobado el presupuesto, el primer punto a trabajar en el proyecto fue definir el proceso productivo que llevaría a cabo la nueva incorporación. Puesto que este consistía en un procedimiento cíclico, había que determinar sus fases y las condiciones para el paso de una a otra, que las gobernaría el controlador del sistema, es decir, el PLC, para su automatización.

Tratando este sistema como un automatismo, se le considera secuencial ya que sus salidas (siguientes estados) dependen tanto de las entradas como de los estados anteriores, en vez de depender únicamente de las entradas como sucede en un automatismo combinacional.

Ya que el automatismo en cuestión era sencillo y contaba con pocos estados, un buen método para representarlo era mediante el uso del GRAFCET (diagrama de control con etapas y transiciones), que serviría además para facilitar la posterior programación del PLC.

### 6.1. Definición GRAFCET

Los GRAFCET son un método gráfico que surgió a partir de las redes de Petri en 1977, a raíz del trabajo de la Asociación Francesa para la Cibernética Económica y Técnica (AFCET). Consisten en una sucesión de etapas, cada una de ellas con acciones asociadas que se ejecutan cuando dicha etapa está activa.

Para activarse una etapa siempre se parte de otra etapa que le sucede. Solamente habrá un cambio de etapas si entre ellas existe una transición y además es válida la receptividad que le corresponde (se cumple cierta condición). Al hecho de que una transición sea válida y su receptividad asociada se cumpla se le conoce como que la transición es franqueable, dando lugar a la desactivación de las etapas previas y activando la posterior.

Las diferentes etapas de un GRAFCET se representan mediante un cuadrado, identificándose cada una con un número en su interior. Además, aquella que está activa se suele representar con un pequeño círculo en su interior, y la etapa inicial, con un doble cuadrado.

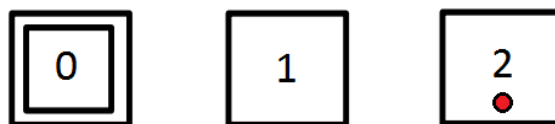
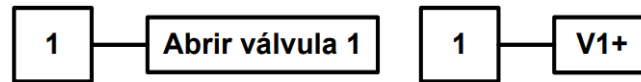


Figura 6.1: Estados de un GRAFCET

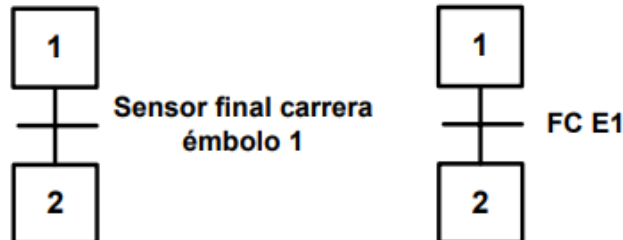
En cuanto a las acciones asociadas, estas se representan con un rectángulo unido al símbolo de su etapa. Para la descripción de estas acciones se utilizan dos niveles:

- **Nivel 1:** Descripción funcional. Se define la acción de una forma global, de modo que pueda entenderse rápidamente su función.
- **Nivel 2:** Descripción operativa. Se detallan las actuaciones que realizará el automatismo (entradas/salidas que se activan).



**Figura 6.2:** Acciones asociadas a un GRAFCET

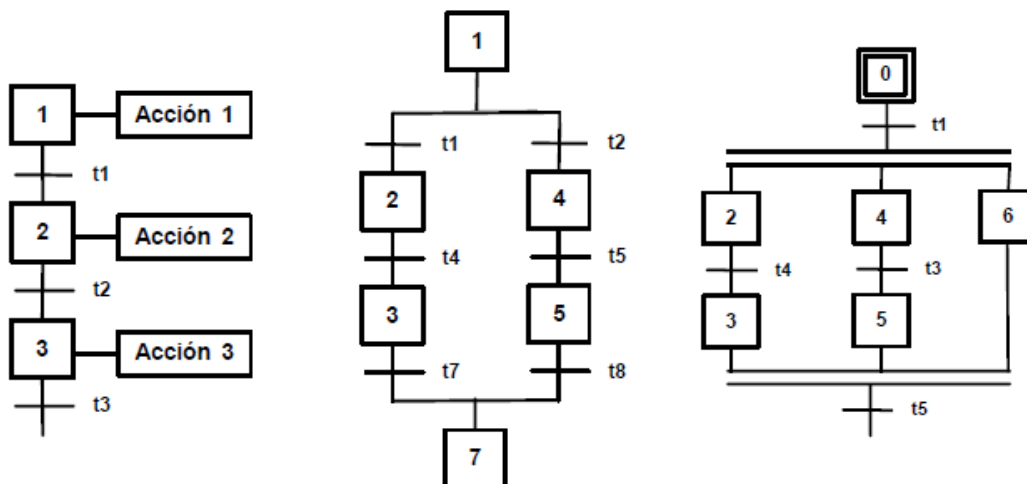
Por otro lado, las transiciones se representan con una línea que une los símbolos de dos etapas, con la receptividad asociada simbolizada con otra línea, perpendicular a la transición y con el texto adjunto de la condición que ha de cumplirse.



**Figura 6.3:** Transiciones y receptividades de un GRAFCET

En referencia a la evolución de los GRAFCET, existen tres estructuras básicas que, al combinarse, permiten representar automatismos más complejos.

- **Secuencia lineal:** Sucesión de etapas con únicamente una transición de llegada y otra de salida.
- **Convergencia o divergencia en “O”:** Presenta alternativas para llegar de una etapa a otra, generalmente siendo secuencias lineales. Únicamente debe cumplirse uno de los caminos. Se puede representar también como una suma “+” de condiciones.
- **Convergencia o divergencia en “Y”:** Presenta distintas condiciones paralelas que deben cumplirse simultáneamente para pasar de una etapa a otra. Se puede representar también como un producto “.” de condiciones.



**Figuras 6.4a, 6.4b y 6.4c:** Secuencia lineal, convergencia en “O” y convergencia en “Y”

Por norma general, las etapas se suelen exponer en vertical y siguiendo un orden descendente, por lo tanto, con las transiciones como líneas verticales y sus receptividades asociadas como líneas horizontales, del mismo modo que el texto. En caso de que la condición para la activación sea que no se cumpla la receptividad escrita, se determina añadiendo “/” al principio del texto.

## 6.2. Guía GEMMA

Ante la complejidad de partir de cero en la programación de un automatismo, es conveniente seguir un procedimiento pautado para evitar olvidarse elementos y tener en cuenta todas las situaciones que pueden darse en su implementación.

Con esta finalidad, la Agencia Nacional francesa para el Desarrollo de la Producción Aplicada a la Industria (ADEPA) diseñó en 1993 la Guía GEMMA. Esta guía se fundamenta en un diseño estructurado que separa el automatismo en tres módulos para facilitar su confección, cada uno con un GRAFCET parcial asociado.

- **Módulo de producción:** Consiste en el proceso básico que gobierna el automatismo en condiciones normales de producción, sin ninguna interferencia. Para ello, se crea el GRAFCET de producción.
- **Módulo de modos de marcha:** Se encarga de dirigir el módulo de producción y efectuar las paradas o puestas en marcha pertinentes según la situación del proceso. Su organigrama se conoce como GRAFCET de conducción.
- **Módulo de seguridad:** Gestiona cómo se procede en casos de emergencias, fallos y defectos. Está a un nivel superior que los dos anteriores ya que puede efectuar paradas sobre los otros módulos sin depender de sus condiciones. En este caso, se le llama GRAFCET de seguridad.

La Guía GEMMA establece una metodología ordenada con los siguientes pasos:

- **Automatización:** En primer lugar, tras una detallada observación del proceso y un estudio de las acciones que deben realizarse, se establece el GRAFCET de primer nivel. Después de tener claras las observaciones, hay que identificar los elementos que serán necesarios para llevar a cabo el automatismo (en este caso las entradas, salidas y conexiones del PLC, además de sensores y actuadores). Finalmente, teniendo ya claros estos elementos, se procede a generar el GRAFCET de segundo nivel. El primer GRAFCET que se genera es el de producción.
- **Supervisión:** Con el organigrama de producción generado, entra en juego la experiencia del operario, ya que hay que reunir el máximo de especificaciones que pueden llegar a darse durante el proceso real del automatismo para poder definir un modo de actuación para cada situación. Con ello surgen los módulos de modos de marcha y de seguridad, por lo que es importante establecer la relación que se existirá entre ellos y también con el de producción, ya creado anteriormente.
- **Interacción:** En relación con la implementación del proceso, es importante definir qué acciones van a depender del operario que vigilará el automatismo. A raíz de esto, hay que diseñar lo que se conoce como Interfaz Hombre-Máquina (HMI) gracias a la cual, el operario interactuará con el automatismo.
- **Implementación:** Con todas las partes ya definidas se puede proceder a programar el automatismo en el lenguaje que precise el PLC.
- **Pruebas:** Por último, la experimentación del automatismo creado será determinante para corregir los errores que se puedan haber dado durante los pasos previos o incluso descubrir elementos que no se hayan tenido en cuenta.

Esta metodología se emplea iterativamente hasta llegar a confeccionar el programa del automatismo y comprobar que funciona correctamente.

### 6.2.1. Representación de la Guía GEMMA

Para facilitar la ejecución de la Guía GEMMA, se definen todas las distintas situaciones que suelen darse en un proceso automatizado, representándose mediante dos diagramas superpuestos.

El primero de ellos, representado mediante cinco rectángulos, distingue cinco situaciones generales, habituales en un automatismo industrial:

- **Control sin alimentar (PZ):** No existencia de energía de alimentación.
- **Procedimientos de parada (A):** Estados de reposo del sistema.
- **Procedimientos de funcionamiento (F):** Estados de producción (ya sea manual o automática).
- **Procedimientos de fallo (D):** Presencia de defectos en la producción y tratamiento en caso de emergencia.
- **Producción:** Estados de producción (engloba tanto la producción normal, como las paradas por mantenimiento, emergencia, etc.).

Las cuatro primeras situaciones se representan con rectángulos grises mientras que la quinta, que engloba partes de la A, F y D, se plasma con un rectángulo superpuesto a los tres anteriores.

El segundo diagrama es una representación más detallada de los distintos modos que pueden darse dentro de cada una de las situaciones A, F y D, expresados con rectángulos menores que se incluyen dentro de la situación en cuestión. Hay un total de 16 modos y no tienen por qué aparecer todos en un proceso, por lo que únicamente se representan los que están en uso. Este diagrama resulta especialmente útil para la fase de supervisión.

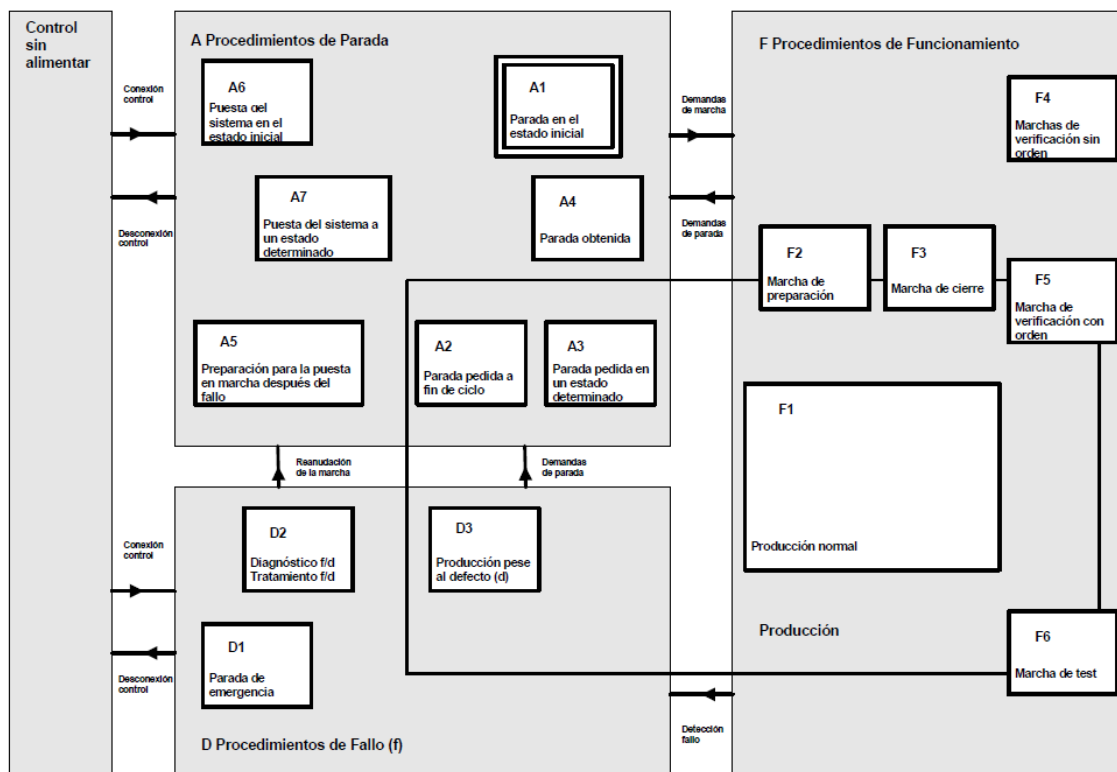


Figura 6.5: Representación de la Guía GEMMA

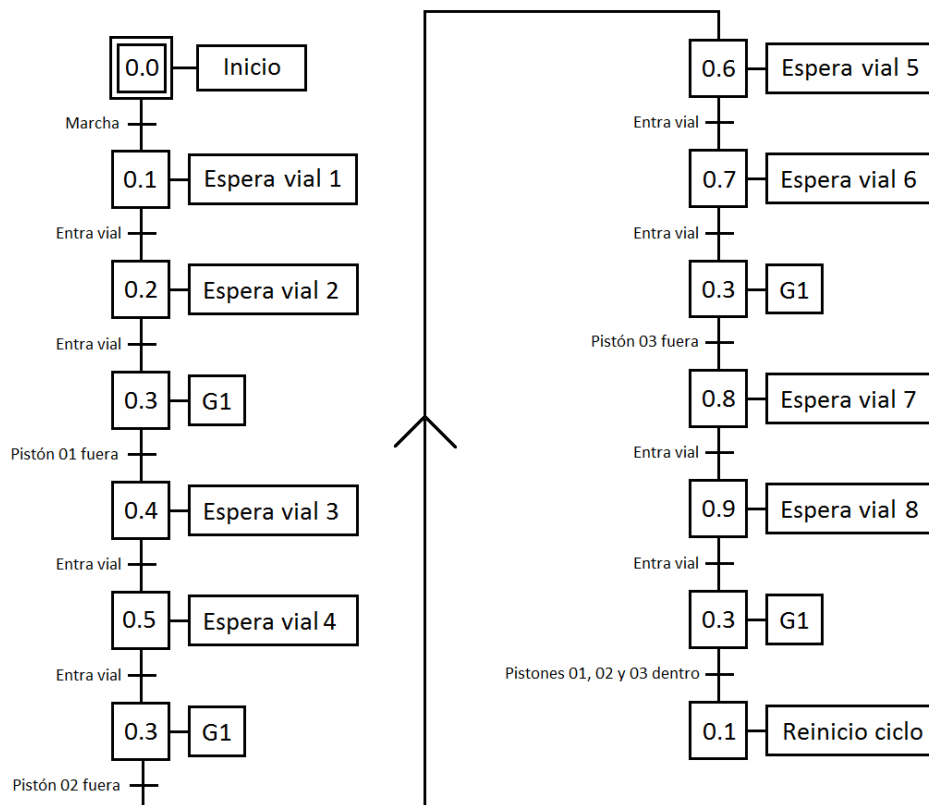
## 6.2.2. Módulo de producción

Siguiendo el orden establecido por la Guía GEMMA, el primer GRAFCET que se generó fue el de producción, en el que se tiene en cuenta solamente el transcurso normal del ciclo de trabajo, sin la interferencia de posibles fallos que se pudiesen dar.

Comenzando con el paso de automatización, había que definir el proceso productivo que se tenía que llevar a cabo y los elementos que lo componían. En este caso, se disponía de una fotocélula en la entrada de los viales a la mesa que realizaría el conteo de estos, tres pistones que servirían como topes para separar las tiras de inyectables, saliendo a medida que otras tres fotocélulas indicasen si las tiras estaban posicionadas. Por último, en cuanto se acabase de rellenar una fila de la mesa, un servo provocaría el avance de la fila para permitir de nuevo el paso de inyectables.

Para una mayor comprensión del proceso, se decidió dividir la producción en dos GRAFCET, separando por un lado la parte de las salidas y entradas de los pistones, y, por otro, el control de la entrada de tiras de inyectables a la mesa.

La representación de estos GRAFCET se muestra en nivel 1 y se les nombra G0 (salida de viales) y G1 (ciclo pistones), siguiendo la Guía GEMMA. Del G0 se hace una llamada al G1 mediante el paso 0.3, quedando parado el primero de ellos mientras se realizan las acciones del G0 según el estado en el que se encuentre el automatismo.



**Figura 6.6:** GRAFCET de producción G0 (salida viales)

En cambio, el GRAFCET G1 no es cíclico, ya que depende del estado de G0. La llamada se representa con la condición X0.3 en referencia al estado de origen.

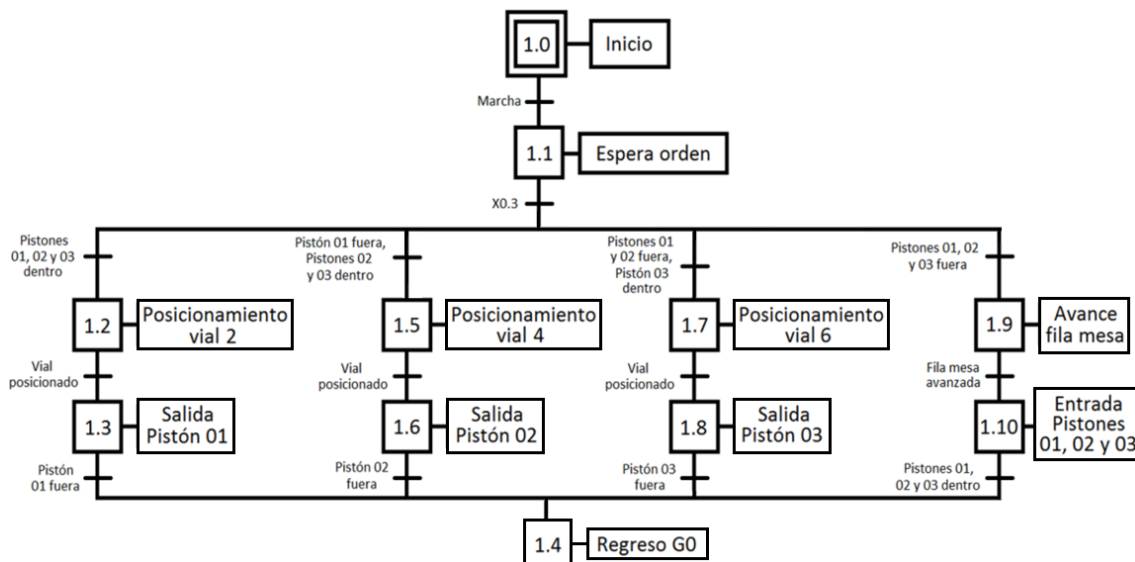


Figura 6.7: GRAFCET de producción G1 (salida/entrada pistones)

### 6.2.3. Módulo de modos de marcha

Con el proceso productivo determinado, era imprescindible definir los modos de parada del ciclo. Para ello se creó el GRAFCET de conducción.

En primer lugar, se definió el procedimiento de petición manual de paro controlado. Se utiliza el modo A3, donde el sistema avanza hasta el siguiente paso en que puede parar y al llegar, se activa el modo A4, que es propiamente la parada. Por otro lado, se incorporó la posibilidad de trabajar en modo manual mediante el modo F5, que requiere de una validación cada vez que se precise avanzar de etapa.

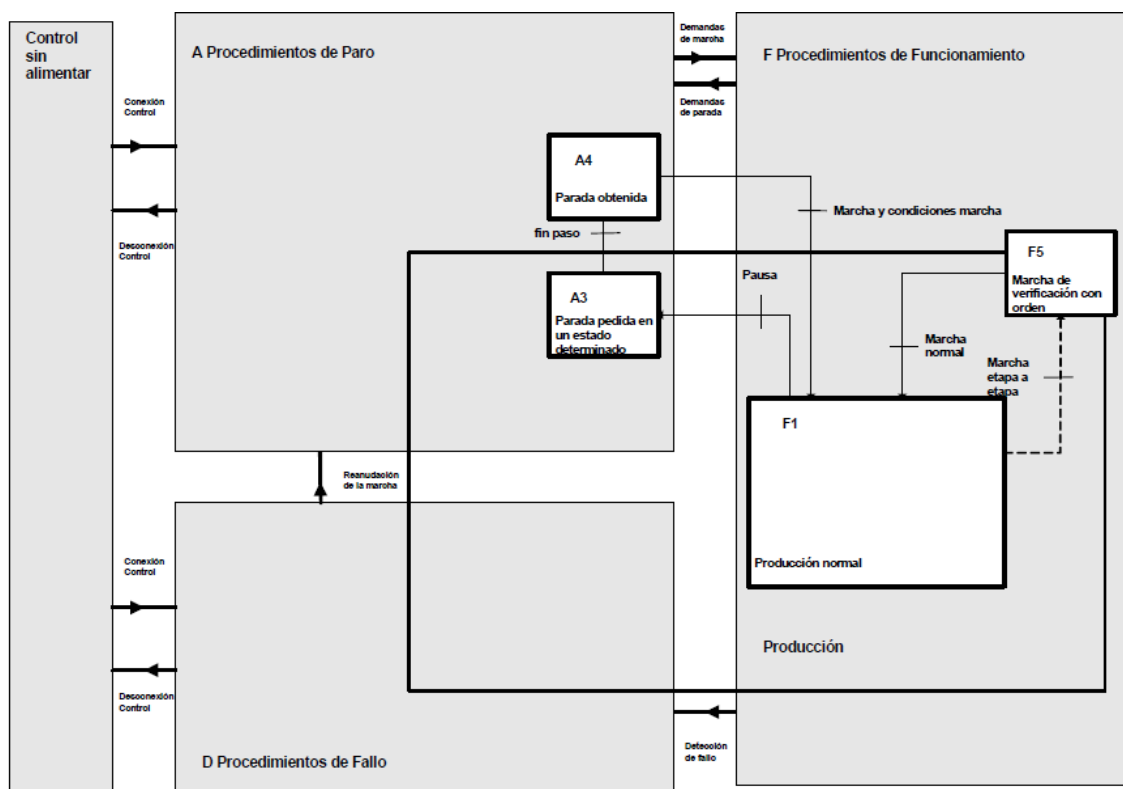


Figura 6.8: Diagrama Guía GEMMA en parada y modo manual

Otro aspecto importante en cuanto a la marcha del ciclo son las actuaciones frente a los diferentes fallos que puedan darse durante la producción. Mediante la supervisión se determinaron las distintas situaciones de fallo que pueden suceder y cómo había que actuar frente a ellas.

- **Vial no llega a su posición en la mesa:** En caso de que una tira de inyectables entre a la mesa y que, al cabo de un determinado tiempo, la fotocélula que verifica su posicionamiento aún no se haya activado, significará que la tira ha quedado encallada en alguna parte de la mesa, por lo que sería necesario parar el ciclo ya que el vial encallado no permitiría el paso de los siguientes. En este caso, el proceso pararía para que el operario pudiese retirar el vial encallado.
- **Presión insuficiente:** El sistema cuenta con un presostato que controla en todo momento que la presión del circuito neumático sea la correcta y hace saltar una alarma en caso contrario. De darse esta alarma, el proceso debería pararse, ya que una falta de presión podría provocar que los pistones no saliesen o entrasen.
- **Pistón no sale/entra:** Si en algún momento, tras accionar la salida o entrada de alguno de los cilindros, este es incapaz de realizar la acción, se podrá saber gracias a los sensores de posición que tienen los cilindros, poniendo un temporizador que cuente el tiempo que tardan en cambiar de posición. Este suceso podría darse por falta de presión, aunque este fallo ya se controla mediante el presostato, por algún fallo mecánico del mismo cilindro o porque alguna tira pueda haber quedado en la posición por donde tiene que salir el pistón. De suceder esto, habría que parar el proceso para que el operario revisase cuál es el motivo. A modo preventivo, el pistón afectado volvería a su posición previa ya que si por ejemplo no puede salir porque un vial entorpece su recorrido, es preferible que vuelva a entrar para no dañarlo.
- **Mesa no avanza:** Aunque este error no dependa propiamente del montaje de los pistones, sí que afecta al transcurso de su ciclo, que quedaría parado hasta que se pudiese solucionar el fallo. No requiere ningún tratamiento especial referente a los pistones.

Tras estas paradas por fallo, siempre el operario debe rearmar el sistema. Debido a que pueden haberse retirado viales de la mesa y el conteo de estos mediante el PLC se vería afectado, siempre que se rearme el sistema tras fallo, se reinicia el ciclo, avanzando una fila de la mesa y entrando los tres pistones para asegurar que el proceso pueda seguir con total normalidad. El hecho de que una fila pueda quedar llena a la mitad no es problemático si son casos puntuales.

Analizándolo mediante la Guía GEMMA, los modos que se usarían para cualquiera de los fallos serían el D2, en el que mediante los contadores o sensores anteriormente mencionados, se identificaría un fallo y se actuaría según estuviese establecido para poder llegar al modo A4, que sería propiamente la parada y que iría acompañada de acciones automáticas predeterminadas para asegurar a los operarios y al proceso. Estando el ciclo parado, el operario actuaría para solucionar el fallo, pasando por el modo A5, en el que se prepararía el proceso para poder reanudar su marcha, que sería siempre desde el inicio del ciclo, ya que mediante el modo A6, se reiniciaría el proceso (rearme), por razones anteriormente explicadas. Por último, para volver a la producción normal, el operario debería activar de nuevo la marcha.





#### 6.2.4. Módulo de seguridad

La Guía GEMMA define el GRAFCET de seguridad para este fin. Debido a que el automatismo estudiado en este proyecto no requería ningún tratamiento específico en caso de darse una emergencia y que, además, el procedimiento no dependía del estado en el que se encontrase el ciclo a la hora de saltar la alarma (simplemente sería necesario parar el ciclo inmediatamente y reiniciarlo al restablecerse el orden), se añadió la función de la seguridad también al GRAFCET de conducción G100.

En cuanto a los modos de la Guía para la formación del GRAFCET, tras accionarse el botón de emergencia se pasa automáticamente al modo D1, parando por completo el ciclo. Después de que se haya solucionado la emergencia, el automatismo prepara el sistema para llevar a cabo un reinicio de ciclo mediante el modo A5. Una vez está el ciclo en su posición inicial (modo A6), el proceso productivo puede restablecerse y volver al modo de producción normal F1.



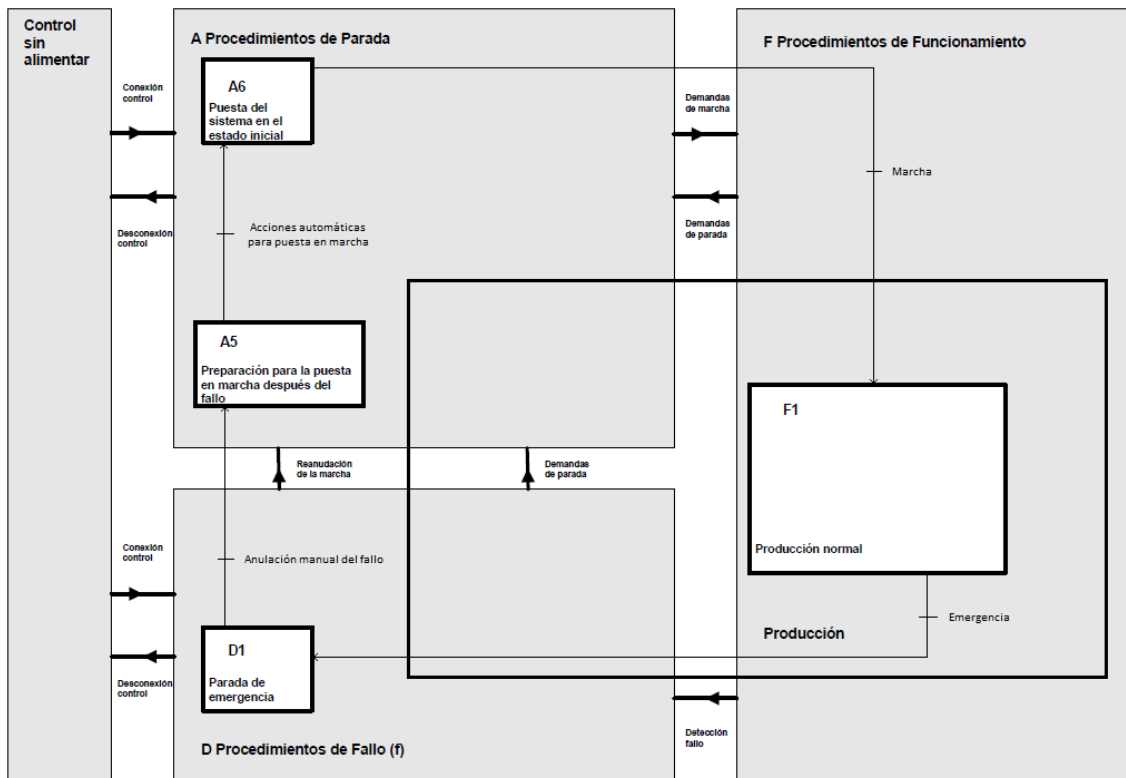


Figura 6.10: Diagrama Guía GEMMA del módulo de seguridad

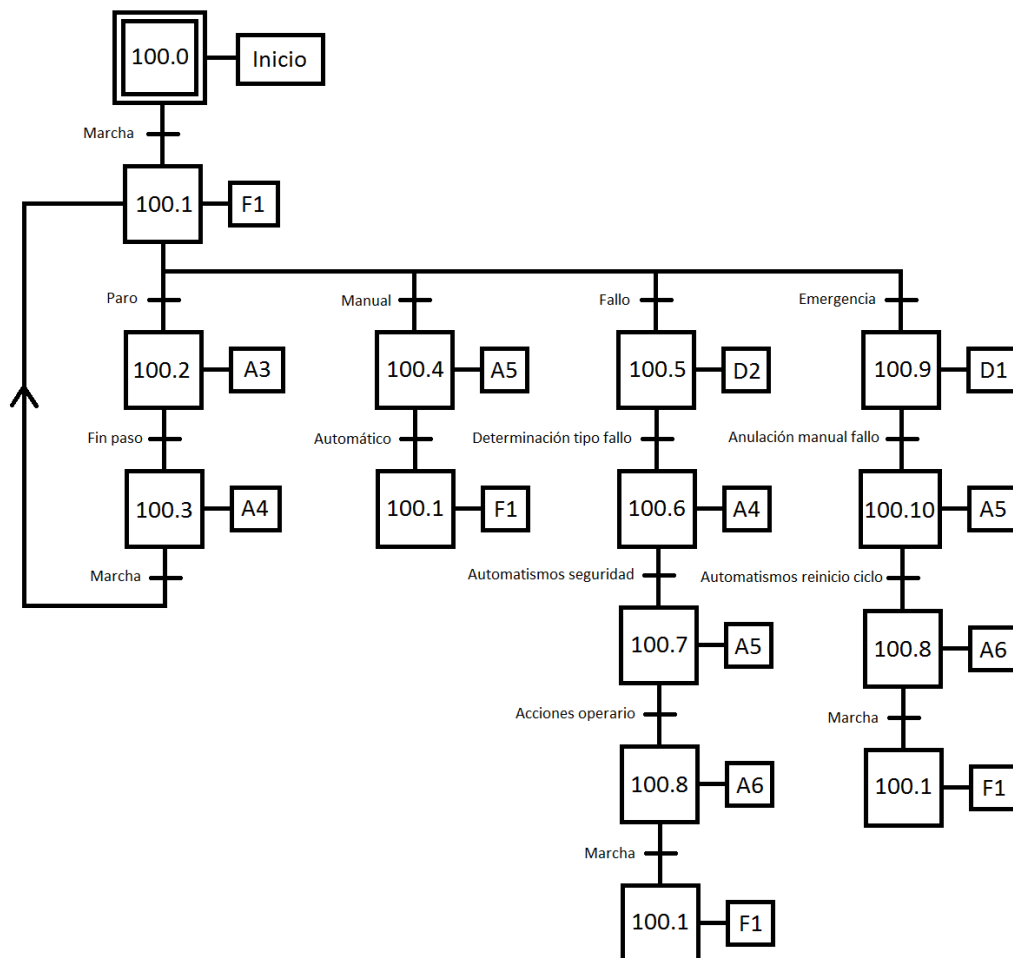


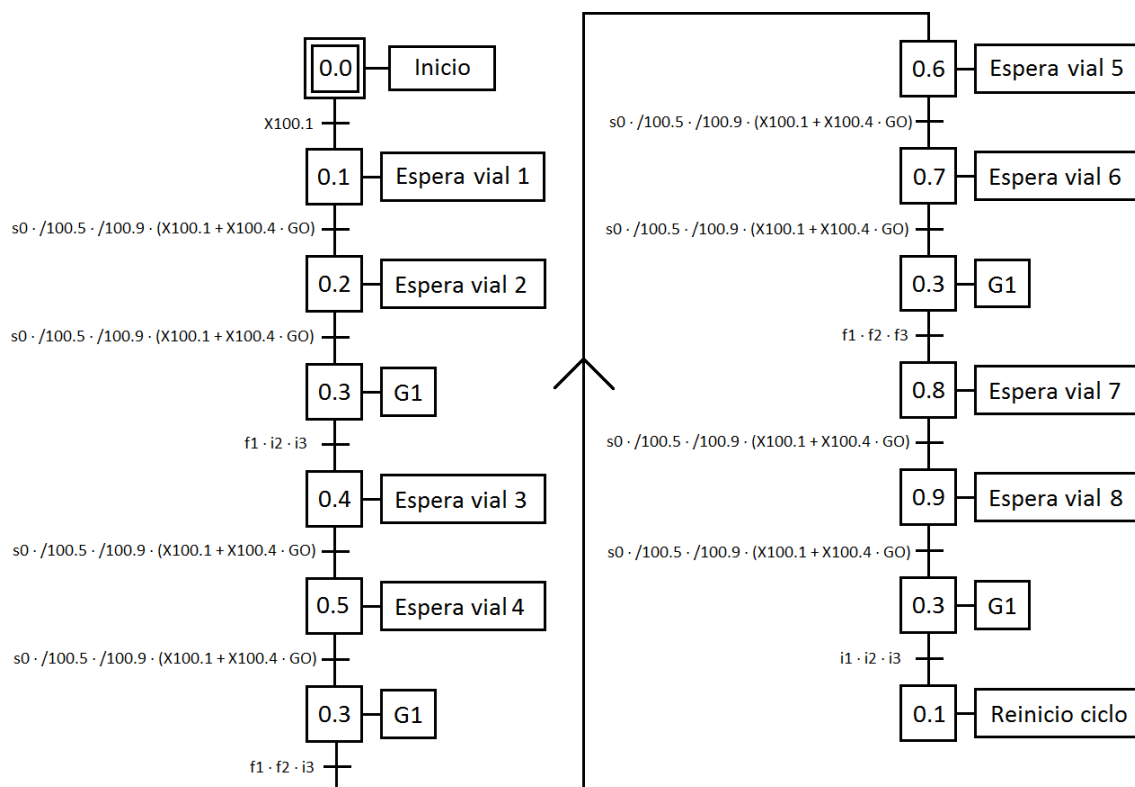
Figura 6.11: GRAFCET de conducción G100

La aparición de nuevos estados en G100 condicionaba los GRAFCET de producción, cuya apariencia final se muestra a continuación en nivel 2. Antes de ello, se especifica la simbología de las entradas y salidas usadas.

**Tabla 6.1:** *Glosario símbolos entradas/salidas GRAFCETS*

Entradas			
s0	Sensor entrada viales mesa	s1	Sensor posición 2ª tira viales mesa
s2	Sensor posición 4ª tira viales mesa	s3	Sensor posición 6ª tira viales mesa
s4	Sensor avance fila mesa	i1	Sensor Pistón 01 dentro
i2	Sensor Pistón 02 dentro	i3	Sensor Pistón 03 dentro
f1	Sensor Pistón 01 fuera	f2	Sensor Pistón 02 fuera
f3	Sensor Pistón 03 fuera	GO	Avanzar paso manual
Salidas			
P1+	Sacar Pistón 01	P2+	Sacar Pistón 02
P3+	Sacar Pistón 03	P1-	Entrar Pistón 01
P2-	Entrar Pistón 02	P3-	Entrar Pistón 03
F+	Avanzar fila mesa		

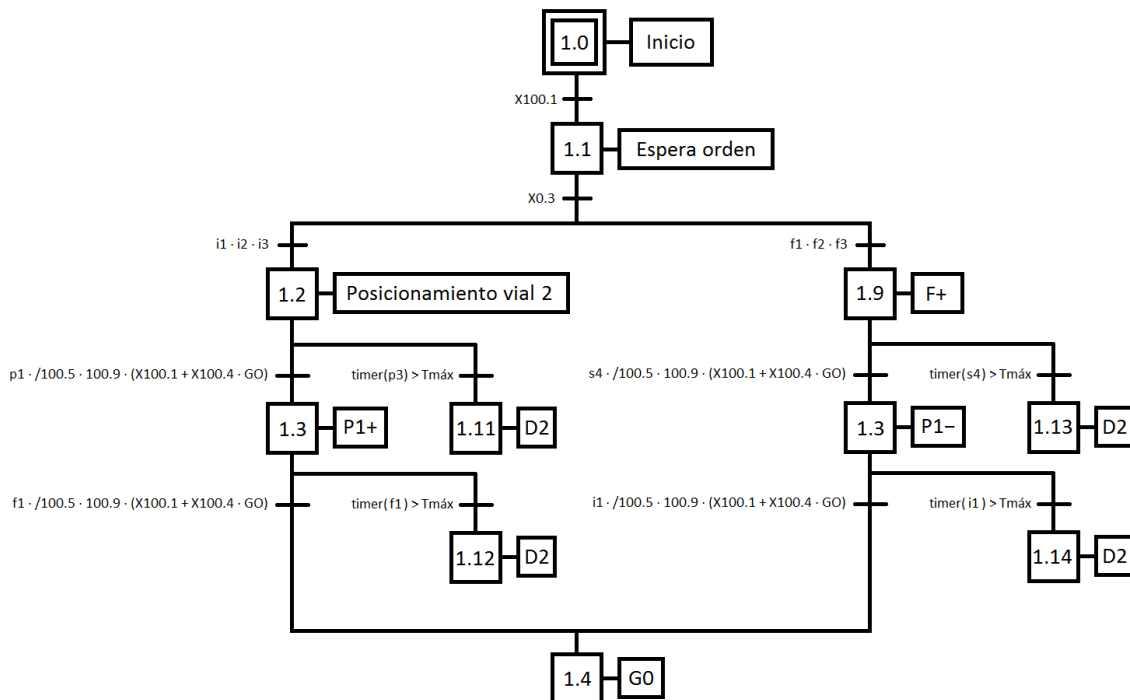
El GRAFCET G0 definitivo se muestra en la *Figura 6.12*:



**Figura 6.12:** *GRAFCET G0 definitivo*

Cabe remarcar que el símbolo “+” equivale a un OR, es decir, debe cumplirse una de las condiciones para poder avanzar el ciclo; el “.” es un AND, donde tienen que cumplirse ambas condiciones para pasar a la siguiente etapa; una “X” frente a un estado significa que ha de estar activo dicho estado para validarse la condición y un “/” corresponde a una condición negada, es decir, no ha de cumplirse para que se pueda avanzar.

Por otro lado, el GRAFCET G1 se representa tomando únicamente uno de los pistones (cualquiera) ya que el ciclo es el mismo para todos ellos. En la *Figura 6.13* se muestra el GRAFCET G1 definitivo para el primer pistón.

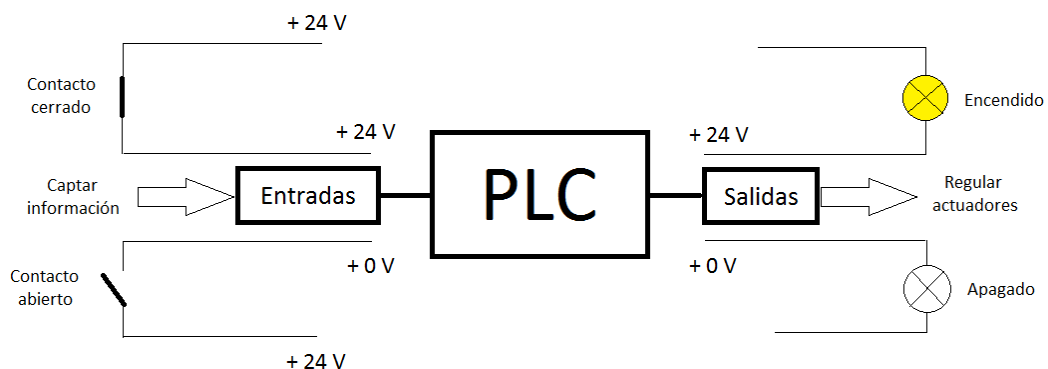


**Figura 6.13:** GRAFCET G1 definitivo

## 7. Controlador Lógico Programable

Un Controlador Lógico Programable es un autómata programable, es decir, una máquina capaz de actuar sobre un sistema en función de su estado y evolución, como podría ser una impresora, un equipo de llenado de tanques de agua o una prensa, gestionado por el programa que incorpora un usuario.

Los PLC cuentan con un sistema de entradas y salidas gracias a los cuales se comunican con el exterior. Tanto las entradas como las salidas son conexiones de por ejemplo 24 V. Las primeras se conectan a generadores de señal, que podrían ser sensores de posición o simples interruptores o pulsadores, mientras que las segundas se conectan a actuadores que, por ejemplo, podrían activar motores o abrir válvulas.



**Figura 7.1:** Esquema básico entradas/salidas PLC

Los PLC están formados por diversas partes que pueden estar integradas o ser modulares. Entre los elementos más importantes se destacan:

- **CPU (Unidad Central de Procesamiento):** Ejecuta el programa que el usuario ha preestablecido, realiza las operaciones aritméticas y lógicas y coordina la comunicación con el resto de componentes. Es el módulo principal del PLC.
- **Fuente de alimentación (PS):** Se encarga de suministrar energía eléctrica a la CPU y al resto de módulos que forman el PLC. Suele encontrarse en un extremo (izquierdo generalmente), justo al lado de la CPU.
- **Módulos de entradas (I) y salidas (O):** Conectan el entorno físico con la CPU del PLC mediante diferentes niveles de tensión. Pueden tratar información analógica (AI/AO) o digital (DI/DO), según la naturaleza del módulo.
- **Módulos de memoria:** Almacenan internamente el programa y cualquier variable necesaria para el automatismo. Pueden ser volátiles (RAM) o no volátiles (ROM). Suelen conectarse directamente al CPU.
- **Puertos de comunicación (CP):** Medio que comunica el PLC con las unidades de programación, otros PLC, Interfaz (HMI), etc.

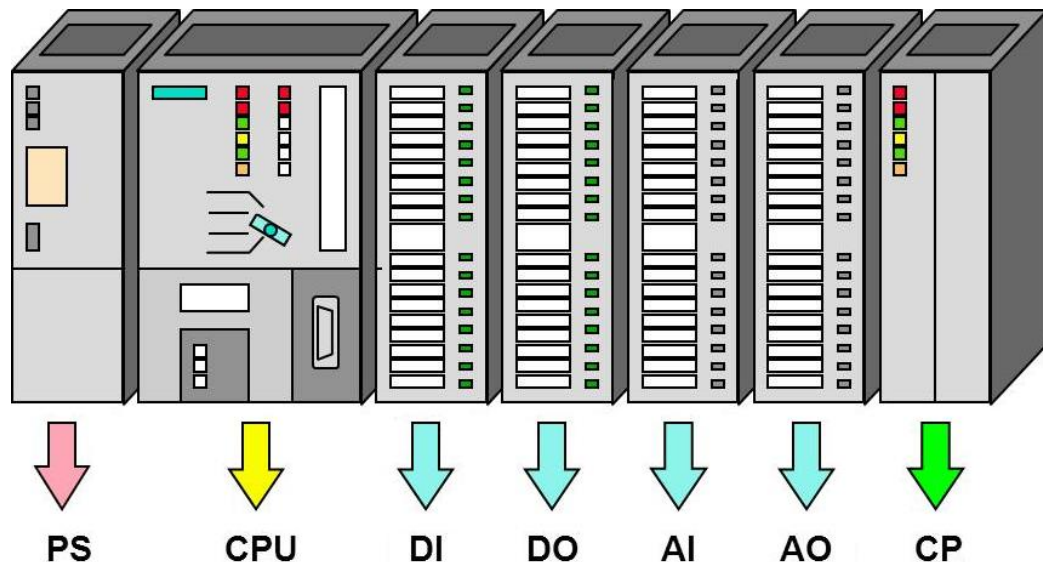


Figura 7.2: Módulos de un PLC

## 7.1. PLC Siemens

Una de las marcas más conocidas de PLC es la alemana Siemens, que cuenta con diversos modelos de la familia SIMATIC S7 según los requerimientos del sistema a controlar. La estación de carga de viales a la que se le va a incorporar la implementación de este proyecto cuenta con un controlador SIMATIC S7-300.

Cabe destacar que cada marca tiene una metodología de programación propia, que aun siendo similares entre sí, requieren cierta formación. Para esta gama de controladores es necesario el uso del entorno de programación SIMATIC STEP7, que permite la creación de proyectos en los que se pueden combinar diversos PLC. Este entorno está pensado simplemente para la elaboración del programa que gobernará el automatismo pero permite incorporar elementos como el HMI (explicado en puntos posteriores).



Figura 7.3: PLC SIMATIC S7-300 de Siemens

### 7.1.1. Lenguajes programación para SIMATIC STEP7

Para el establecimiento de la secuencia ordenada de instrucciones que debe realizar el automatismo existen diversos lenguajes de programación, es decir, conjuntos de símbolos y textos inteligibles por la unidad de programación. En el caso del entorno de programación SIMATIC STEP7 para los autómatas programables de Siemens, existen las siguientes opciones:

- **AWL (lista de instrucciones):** Consiste en un lenguaje de texto en el que se citan por líneas las diferentes operaciones. Este es el lenguaje más completo, ya que permite programar todas las operaciones que el automatismo es capaz de realizar, mientras que con los demás lenguajes existen limitaciones. Todos los otros lenguajes pueden traducirse a AWL pero hay funciones de AWL que no existen en los otros lenguajes. Además, permite añadir comentarios al código para facilitar su lectura por parte del usuario, que no afectan al transcurso del programa ya que son ignorados por el PLC. Cabe destacar que los símbolos son referentes a las expresiones en alemán.
- **KOP (diagrama de contactos):** Se trata de la representación del programa mediante el esquema de contactos o *Ladder*, que asemejaría el automatismo realizado eléctricamente. Es por ello que resulta más sencillo de seguir para usuarios no expertos o usuarios habituados al ámbito eléctrico.
- **FUP (diagrama de funciones):** Utiliza cuadros del álgebra booleana. Resulta especialmente útil para operaciones booleanas en serie por ser compacto y sencillo de utilizar.
- **SCL (lenguaje de texto estructurado):** Es un paquete opcional que cuenta con un lenguaje de alto nivel semejante al *Pascal*. Suele ser útil para usuarios familiarizados con programación de Python o similares.
- **GRAFCET (lenguaje funcional de secuencia):** Es también un paquete opcional que permite programar automatismos cíclicos, partiendo de su diagrama de estados.

En el programa SIMATIC STEP7 de la empresa Fresenius Kabi SAU únicamente se contaba con la opción de usar los tres primeros, por lo que se combinó el uso del FUP para las operaciones que requiriesen enlazar múltiples condiciones booleanas y para el resto, el AWL.

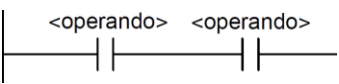
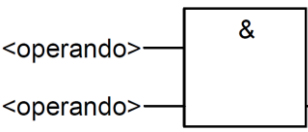
## 7.1.2. Instrucciones básicas

Debido a que los lenguajes de programación mostrados en el anterior apartado cuentan con múltiples instrucciones básicas, en este punto se exponen las principales y más utilizadas a lo largo del proyecto, diferenciando sus diferentes formas según el lenguaje utilizado.

### Operaciones básicas con bits:

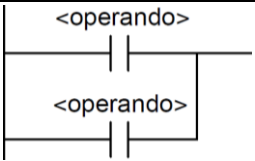
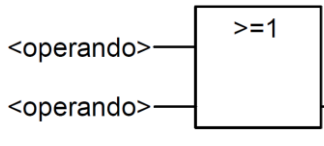
- **UND (Y):** Consulta el bit direccionado y activa la operación si dicho bit es “1”. Si hay más de una condición en serie, todas ellas deben ser “1” para que el resultado de la operación también lo sea. Sería equivalente a contactos en serie en idioma KOP.

**Tabla 7.1:** Símbolos de UND en los diferentes lenguajes

AWL	KOP	FUP
U <operando>		

- **ODER (O):** El resultado de la operación es “1” si al menos una de las condiciones es “1”. Sería equivalente a contactos en paralelo en idioma KOP.

**Tabla 7.2:** Símbolos de ODER en los diferentes lenguajes

AWL	KOP	FUP
O <operando>		

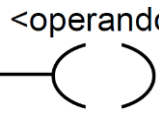
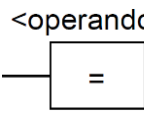
- **NOT (No):** Invierte la condición de entrada, por lo que siempre que la entrada sea un “1”, la salida será un “0” y viceversa. También se puede añadir a otras funciones como la UND (obteniendo UN), a la ODER (ON), etc. Sería el equivalente a un contacto normalmente cerrado en lenguaje KOP. En lenguaje FUP se representa la negación de una entrada o salida mediante un punto.

**Tabla 7.3:** Símbolos de NOT en los diferentes lenguajes

AWL	KOP	FUP
NOT <operando>		

- **Asignar:** Escribe a la salida en cuestión el valor binario de la entrada. Sería equivalente a una bobina de relé cuando se representa en lenguaje KOP.

**Tabla 7.4:** Símbolos de Asignar en los diferentes lenguajes

AWL	KOP	FUP
= <operando>		

- **SET (Activar):** Escribe el valor “1” al bit de salida siempre que el RLO (condiciones de activación) sea igual a “1”. La salida permanece a “1” hasta que no se vuelva a actuar sobre ella para desactivarla.

**Tabla 7.5:** Símbolos de SET en los diferentes lenguajes

AWL	KOP	FUP
S <operando>	<operando> —(S)	<operando> —[S]

- **RESET (Desactivar):** Escribe el valor “0” al bit de salida siempre que el RLO (condiciones de activación) sea igual a “0”. La salida permanece a “0” hasta que no se vuelva a actuar sobre ella para activarla.

**Tabla 7.6:** Símbolos de RESET en los diferentes lenguajes

AWL	KOP	FUP
R <operando>	<operando> —(R)	<operando> —[R]

- **Flanco positivo:** Si detecta que hay un cambio de “0” a “1” en el RLO, da una señal de “1” en su salida. De lo contrario, permanece a “0”. Requiere un área de memoria que recuerde el estado anterior de la entrada, ya que, para existir un flanco positivo, debe ser el estado anterior (operando 2) un “0” y el actual (operando 1) un “1”. Existe también un Flanco negativo que actúa del mismo modo.

**Tabla 7.7:** Símbolos de Flanco positivo en los diferentes lenguajes

AWL	KOP	FUP
U <operando1> FP <operando2>	<operando1> — POS Q <operando2>— M_BIT	<operando1> — POS <operando2>— M_BIT Q

#### Operaciones de comparación y aritmética:

En la programación de Siemens se distinguen entre números enteros (I) de 16 bits, enteros dobles (D) de 32 bits y números en coma flotante (R), de 32 bits. Las operaciones son las mismas, pero siempre hay que indicar de qué tipo es la operación (I, D o R).

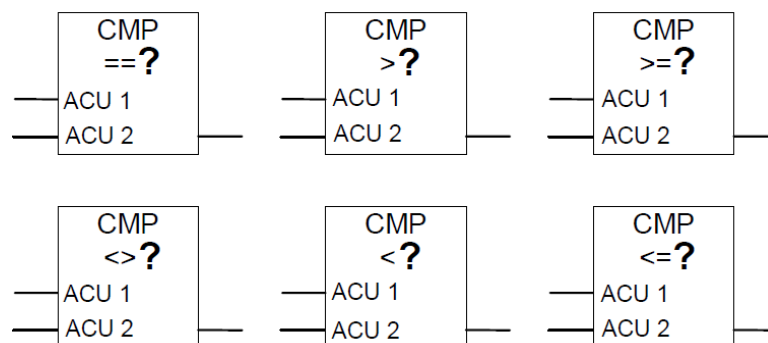
- **Comparación:** En idioma AWL, se comparan las dos líneas superiores de código antes de expresar la función, que se expresan como ACU1 y ACU2 a medida que aparecen en el código y su salida es “1” si la comparación es cierta y “0” de lo contrario. El tipo de número se expresa sustituyendo el “?” del ejemplo por I, D o R. Las funciones posibles se muestran en la *Tabla 7.8*.

**Tabla 7.8:** Operadores de comparación en AWL

==?	ACU1 igual que ACU2	<>?	ACU1 diferente que ACU2
>?	ACU1 mayor que ACU2	<?	ACU1 menor que ACU2
>=?	ACU1 mayor o igual que ACU2	<=?	ACU1 menor o igual que ACU2



En cuanto a los lenguajes KOP y FUP, la metodología es la misma y ambos tienen los mismos símbolos.



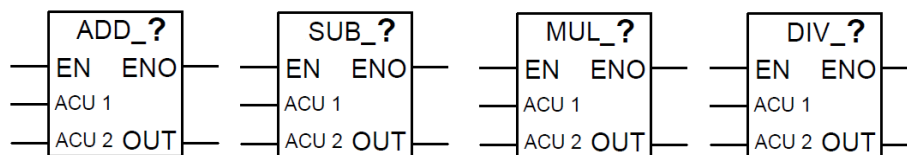
**Figura 7.4:** Símbolos operadores comparación KOP y FUP

- **Aritmética:** Las operaciones aritméticas se ejecutan del mismo modo que las de comparación.

**Tabla 7.9:** Operadores de aritmética en AWL

+	Sumar ACU1 y ACU2	-	Restar ACU1 y ACU2
*	Multiplicar ACU1 y ACU2	/	Dividir ACU1 y ACU2

En cuanto a los lenguajes KOP y FUP, los símbolos son también iguales.



**Figura 7.5** Símbolos operadores comparación KOP y FUP

#### Operaciones de salto:

Cabe la posibilidad de interrumpir el desarrollo normal del programa si se dan ciertas condiciones. Los más utilizados son el SPB y el SPBN que se activan si el RLO es “1” o “0” respectivamente. Estas operaciones únicamente pueden utilizarse con el lenguaje AWL. Un ejemplo de uso sería:

U <operando>  
SPBN <acción 1>

JOVR: <acción 2>

En este caso, si el operando es igual a “1”, se llevará a cabo la acción 1 y posteriormente se seguirá con el transcurso normal del programa. De lo contrario, se saltará hasta la meta “JOVR” (puede nombrarse de cualquier manera siempre y cuando sean cuatro caracteres y no se repitan dentro de un mismo programa para evitar problemas), sin realizarse la acción 1.

#### Operaciones de carga y transferencia

En ocasiones es necesario intercambiar información entre los módulos de entradas y salidas con los de memoria o dentro de los mismos módulos de memoria entre sí.



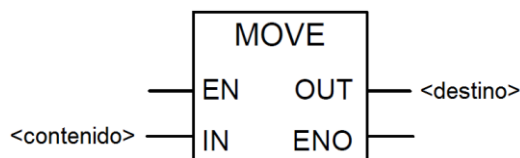
Estas operaciones son incondicionales, es decir, se producen en cada ciclo sin depender de ningún resultado lógico. Es por esto que resulta útil combinarlas con las operaciones de salto si se desea evitar su uso en condiciones determinadas.

El contenido que se trata con estas operaciones puede ser bytes (8 bits), palabras o words (16 bits) o dobles palabras o double words (32 bits). La forma de tratarlas es la misma para los tres tipos de contenido citados.

Si se utiliza el lenguaje AWL, es necesario utilizar dos instrucciones: Load (Cargar) representado con una L y Transfer (Transferir), representado con una T.

L <contenido>  
T <destino>

En cuanto a los lenguajes KOP y FUP, se puede realizar esta operación de transferencia de datos mediante un único bloque, igual en los dos lenguajes.



**Figura 7.6:** Bloque de transferencia lenguajes KOP y FUP

#### Operaciones de conteo:

La CPU tiene reservada una parte de memoria para palabras de 16 bits que ejercen como contadores, con un máximo de 256 de ellos. Los contadores se nombran con la letra Z acompañada del número que lo identifica (Z???). Las operaciones básicas son las siguientes:

- **Poner contador en el valor inicial:** Carga el valor del ACU1 si se activa el RLO. El ACU1 debe estar en formato BCD (C#???) y estar entre el 0 y el 999.

**Tabla 7.10:** Símbolos de Poner contador en valor inicial en los diferentes lenguajes

AWL	KOP	FUP
U <operando> L <valor> S <contador>		

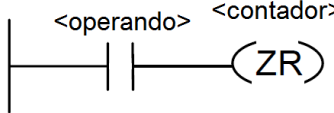
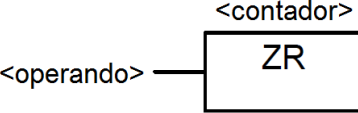
- **Incrementar contador:** Aumenta en “1” el valor del contador si se activa el RLO. En caso de llegar al 999, el contador dejará de aumentar y el valor del RLO no tendrá efecto alguno. No se activa el bit de desbordamiento (OV).

**Tabla 7.11:** Símbolos de Incrementar contador en los diferentes lenguajes

AWL	KOP	FUP
U <operando> ZV <contador>		

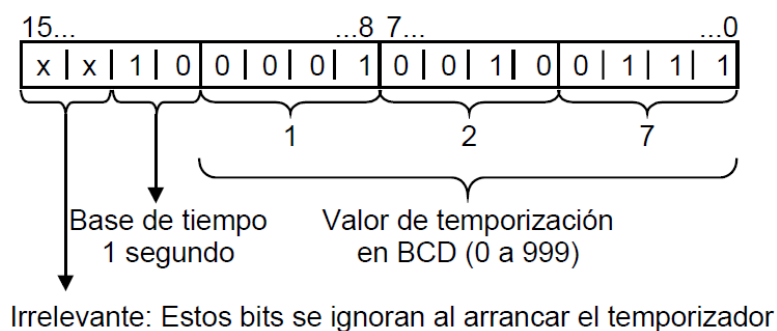
- **Decrementar contador:** Disminuye en “1” el valor del contador si se activa el RLO. En caso de llegar al 0, el contador dejará de disminuir ya que no se puede trabajar con números negativos, por lo que el valor del RLO no tendrá ningún efecto.

**Tabla 7.12:** Símbolos de Decrementar contador en los diferentes lenguajes

AWL	KOP	FUP
U <operando> ZR <contador>		

#### Operaciones de temporización:

Los temporizadores, como los contadores, disponen de un área reservada de memoria de la CPU, que cuenta con una palabra de 16 bits para cada uno de ellos, con un máximo de 256. Los temporizadores se nombran mediante la letra T seguida del número que lo identifica (T???). Los bits del 0 al 11 de la palabra que representa el temporizador contienen su valor en BCD. Los bits 12 y 13 almacenan la información en binario de la base de tiempo.



**Figura 7.7:** Disposición bits en palabras de temporización

El orden de temporización es descendente, es decir, finaliza cuando se llega a 0. El valor inicial de tiempo se puede cargar en distintos formatos: binario, hexadecimal o BCD y puede ir de 0 a 9990 segundos (o 2 horas, 46 minutos y 30 segundos). Para cargar el valor de temporización pueden usarse los siguientes formatos:

- **w#16#txyz:** Siendo “t” la base de tiempo y “xyz” el valor en BCD.
- **S5T#aH\_bM\_cS\_dMS:** Siendo “a” el valor de horas, “b” el de minutos, “c” el de segundos y “d” el de milisegundos, asignándose la base de tiempo automáticamente.

La base de tiempo es el intervalo que se decrementa cada vez que se actualiza el temporizador, codificadas de la forma mostrada en la *Tabla 7.13*.

**Tabla 7.13:** Codificación bases de tiempo

Código	Base de tiempo
00	10 ms
01	100 ms
10	1 s
11	10 s

Los valores de tiempo se almacenan con una única base de tiempo, por lo que, si no son exactamente múltiplos del intervalo, se truncan. Si la resolución es demasiado alta, se redondean siguiendo el siguiente criterio de la *Tabla 7.14*.

**Tabla 7.14:** Resolución en función del valor temporal

Resolución	Intervalo de tiempo
0,01 segundos	[10MS, 9S_990MS]
0,1 segundos	[100MS, 1M_39S_900MS]
1 segundo	[1S, 16M_39S]
10 segundos	[10S, 2H_46M_30S]

Existen cinco tipos de temporizadores con distinto comportamiento frente al RLO que los acciona:

- **S\_IMPULS (Temporizador de impulso):** Mientras esté activada la entrada del temporizador, su salida permanecerá en “1” el tiempo que se le asigne desde el momento de la activación. Si se desactiva el temporizador, aunque no haya pasado el tiempo establecido, el temporizador volverá a “0”. En los lenguajes KOP y FUP, el bloque es el mismo.

**Tabla 7.15:** Símbolos de S\_IMPULS en los diferentes lenguajes

AWL	KOP y FUP
U <operando> L <valor> SI <temporizador>	<div style="text-align: center;">&lt;temporizador&gt;</div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">&lt;operando&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SI</div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="margin-right: 10px;">&lt;valor&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">TW</div> </div>

La salida se llama mediante el nombre “T??”, que actúa como un bit.

- **S\_VIMP (Temporizador de impulso prolongado):** La señal de salida permanece en “1” durante el tiempo establecido independientemente de la señal de activación.

**Tabla 7.16:** Símbolos de S\_VIMP en los diferentes lenguajes

AWL	KOP y FUP
U <operando> L <valor> SV <temporizador>	<div style="text-align: center;">&lt;temporizador&gt;</div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">&lt;operando&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SV</div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="margin-right: 10px;">&lt;valor&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">TW</div> </div>

- **S\_EVERZ (Temporizador de retardo a la conexión):** La señal de salida es “1” si ha finalizado el tiempo programado y la señal de activación sigue siendo “1”, hasta que deja de serlo.

**Tabla 7.17:** Símbolos de S\_EVERZ en los diferentes lenguajes

AWL	KOP y FUP
U <operando> L <valor> SE <temporizador>	<div style="text-align: center;">&lt;temporizador&gt;</div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">&lt;operando&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SE</div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="margin-right: 10px;">&lt;valor&gt; —</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">TW</div> </div>

- **S\_SEVERZ (Temporizador de retardo a la conexión con memoria):** La señal de salida se activa cuando ha pasado el tiempo programado, sin depender del estado de la señal de activación, que solo actúa para iniciar el temporizador.

**Tabla 7.18:** Símbolos de S\_SEVERZ en los diferentes lenguajes

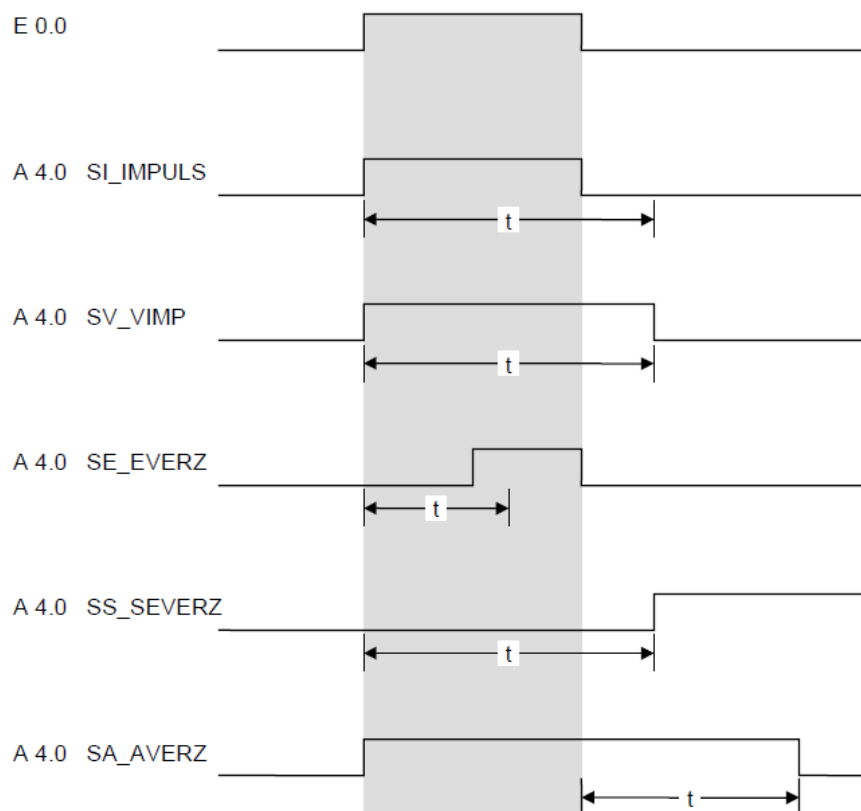
AWL	KOP y FUP
U <operando> L <valor> SS <temporizador>	<div style="text-align: center;">&lt;temporizador&gt;</div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: right; margin-right: 10px;">&lt;operando&gt;</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SS</div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="text-align: right; margin-right: 10px;">&lt;valor&gt;</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">TW</div> </div>

- **S\_AVERZ (Temporizador de retardo a la desconexión):** La señal de salida es "1" mientras lo sea la de entrada y lo continúa siendo al desactivarse la entrada, durante el tiempo programado.

**Tabla 7.19:** Símbolos de S\_AVERZ en los diferentes lenguajes

AWL	KOP y FUP
U <operando> L <valor> SA <temporizador>	<div style="text-align: center;">&lt;temporizador&gt;</div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: right; margin-right: 10px;">&lt;operando&gt;</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SA</div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 10px;"> <div style="text-align: right; margin-right: 10px;">&lt;valor&gt;</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">TW</div> </div>

En la *Figura 7.8* se muestran de forma resumida las características principales de cada uno de los tipos de temporizadores.



**Figura 7.8:** Resumen del tipo de temporizadores

### 7.1.3. Módulos y áreas de memoria

A la hora de programar un PLC, es esencial hacerlo de forma ordenada para facilitar el proceso y sobre todo, para posteriores revisiones del programa, ya que este debe ser entendible. Para ello, STEP7 cuenta con la posibilidad de emplear distintos módulos o bloques dentro de un mismo programa para estructurar el código.

- **Módulos de organización (OB):** Representan la interface entre el sistema operativo de la CPU y el programa proporcionado por el usuario. Son los módulos principales ya que se encargan de llamar al resto y tienen la capacidad de parar el programa si es necesario. Hay diversos OB, cada uno identificado por un número ya establecido por el entorno de programación, con distintas utilidades. De entre ellos se destacan el OB1, que se encarga de ejecutar cíclicamente el programa; los OB100-102, que son encargados del arranque del PLC; y los OB30-38, que son cíclicos, es decir, actúan rigiéndose por un reloj en vez de depender el ciclo del programa. Además hay otros encargados de las alarmas, fallos, etc.
- **Funciones (FC):** Suelen ser llamadas desde los OB (aunque también pueden llamarse desde otras FC o de FB) y contienen propiamente el programa para evitar tener que añadirlo en el OB, dedicándolo así únicamente para la gestión del resto de módulos. Generalmente son usadas para subrutinas que no requieran un almacenaje de datos, como podrían ser ejecuciones de ciclos, cálculos, etc.
- **Bloques de función (FB):** Son similares a las FC pero con la capacidad de almacenar datos, ya que van siempre asociados a un DB de instancia (se define posteriormente). Esta característica resulta muy útil en instalaciones con diversos elementos que se encuentran repetidos y que realizan un ciclo parecido, como podrían ser los tres cilindros neumáticos en el caso estudiado en este proyecto, ya que se establece un mismo ciclo para todos pero con distintas variables (tiempos de ejecución, condiciones, etc.) para distinguirlos, usando un FB para todos pero con un DB asociado a cada uno de los cilindros.
- **Módulos de datos (DB):** Son áreas de memoria que actualiza el usuario pero permanecen durante los ciclos del programa. Pueden ser globales, cuando se usan simplemente para almacenar datos que luego pueden ser llamados desde un OB, FC o FB o si no, pueden ser de instancia, que siempre van asociados a un FB y almacenan los datos de esa subrutina en cuestión.
- **Módulos de sistema (SFC/SFB):** Al igual que las FC y los FB, existen una serie de bloques predeterminados por Siemens. Tal y como los FB, los SFB requieren de una SDB de instancia. Ejemplos de ello son el SFB4, que es un temporizador TON (retardo a la conexión) o el SFB5, que es un TOF (temporizador con retardo a la desconexión).

Todos los módulos anteriormente citados están estructurados por segmentos, que sirven para separar las distintas acciones que se realizan dentro de un mismo programa.

Además de estos bloques, es importante destacar la existencia de las Marcas, que son bits internos de la CPU y pueden usarse a modo de memoria como los DB.

## 7.2. Programa del PLC

### 7.2.1. Ocupación de variables

Al tratarse de un PLC que ya contaba con un programa en funcionamiento, primero había que analizar el programa existente y adaptarlo para el nuevo ciclo, añadiendo las partes necesarias.

Una vez claro el punto de partida, lo siguiente fue averiguar qué entradas y salidas estaban libres para poder usarse en el nuevo programa, al igual que las marcas de memoria, temporizadores y contadores. En cuanto a las entradas y salidas, esto podía comprobarse mediante los esquemas eléctricos o mediante el mismo STEP7. En la *Figura 7.9* muestra cómo se pueden ver las salidas libres del PLC.

/	7	6	5	4	3	2	1	0
EB 0	X	X	X	X	X	X	X	X
EB 1	X	X	X	X	X	X	X	X
EB 2	X	X	X	X	X	X	X	X
EB 3	X	X	X	X	X	X	X	X
EB 4	X	X	X	X	X	X	X	X
EB 5	X	X	X	X	X	X	X	X
EB 6	X	X	X	X	X	X	X	X
EB 7						X	X	X
AB 0	X	X		X		X		X
AB 1	X	X	X	X	X	X	X	X
AB 2	X	X	X	X	X	X	X	X
AB 3	X	X	X	X	X	X	X	X
AB 4		X	X	X	X	X	X	X
AB 5	X				X	X	X	X
AB 6	X							

**Figura 7.9:** Ocupación entradas y salidas del programa previo

Las entradas y salidas son grupos de bytes que se representan con una letra según sean entradas (E) o salidas (A) por su nombre en alemán, seguidas del número de byte que representan, un punto y la posición del bit. Un ejemplo de salida podría ser A2.4 y otro de entrada, E5.0. En la *Figura 7.9* se pueden ver por filas los bytes y por columnas los bits, creando una tabla con una “X” si la entrada o salida está ocupada.

Para el problema se requerían nueve entradas (dos por cada pistón para los detectores magnéticos y tres más para las fotocélulas de posicionamiento de viales en la mesa) y seis salidas (dos por cada pistón ya que son biestables, es decir, tienen dos posiciones y cada acción de entrar y salir va por separado). Por cuestiones de orden, lo mejor era situar las entradas y salidas nuevas en las últimas posiciones. En cuanto a las salidas, el byte 6 aparecía libre, por lo que podían usarse sus seis primeros bits, quedando de la forma expresada en la *Tabla 7.20*.

**Tabla 7.20:** Símbolos salidas

A6.0	O_Piston01Salir	A6.1	O_Piston01Entrar
A6.2	O_Piston02Salir	A6.3	O_Piston02Entrar
A6.4	O_Piston03Salir	A6.5	O_Piston03Entrar

También por estructurar el programa, todas las salidas (y el resto de variables) se codificaron del mismo modo: con una “O” las salidas (por el nombre inglés), una “I” para las entradas o una “M” para las marcas.

En el caso de las entradas, según la *Figura 7.9* parece que estén ya ocupadas la mayoría, pero esto es porque simplemente aparecen las entradas o salidas que tienen algún bit ocupado, por lo que había que mirar si en el hardware habían módulos de entradas disponibles. Esto también podía mirarse desde el STEP7, tal y como se muestra en la *Figura 7.10*.

Slot	Módulo	Referencia	Firmware	Dirección MPI	Dirección E	Dirección S
4	PM-E DC24V	6ES7 138-4CA01-0AA0			2040*	
5	8DI DC24V	6ES7 131-4BF00-0AA0			0.0...0.7	
6	8DI DC24V	6ES7 131-4BF00-0AA0			1.0...1.7	
7	8DI DC24V	6ES7 131-4BF00-0AA0			2.0...2.7	
8	8DI DC24V	6ES7 131-4BF00-0AA0			3.0...3.7	
9	8DI DC24V	6ES7 131-4BF00-0AA0			4.0...4.7	
10	8DI DC24V	6ES7 131-4BF00-0AA0			5.0...5.7	
11	8DI DC24V	6ES7 131-4BF00-0AA0			6.0...6.7	
12	8DI DC24V	6ES7 131-4BF00-0AA0			7.0...7.7	
13	8DI DC24V	6ES7 131-4BF00-0AA0			8.0...8.7	
14	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				0.0...0.7
15	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				1.0...1.7
16	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				2.0...2.7
17	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				3.0...3.7
18	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				4.0...4.7
19	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				5.0...5.7
20	8DO DC24V/0.5A	6ES7 132-4BF00-0AA0				6.0...6.7

**Figura 7.10:** Módulos instalados por hardware

Se puede comprobar que había instalado otro módulo de entradas, es decir, no había problema de espacio. Los símbolos de las entradas se exponen en la *Tabla 7.21*.

**Tabla 7.21:** Símbolos entradas

E7.3	I_Piston01Dentro	E7.4	I_Piston01Fuera
E7.5	I_Piston02Dentro	E7.6	I_Piston02Fuera
E7.7	I_Piston03Dentro	E8.0	I_Piston03Fuera
E8.1	I_Vial02Posicionado	E8.2	I_Vial04Posicionado
E8.3	I_Vial06Posicionado		

Para las marcas, la manera de actuar fue la misma. Estas se representan del mismo modo que las entradas y salidas, con una "M". Se decidió usar a partir de la 170 por cuestiones también de estructurar el programa.

	7	6	5	4	3	2	1	0	B	W	D
MB170											
MB171											
MB172											
MB173											
MB174											
MB175											
MB176											
MB177											
MB178											
MB179											
MB180											

**Figura 7.11:** Ocupación marcas

Las marcas se usarían para diversas aplicaciones como indicar que el pistón está listo para salir, que hay un estado de alarma, etc. Es importante el hecho de codificar las distintas variables con símbolos (los nombres determinados por el usuario) para agilizar su uso, aunque también pueden usarse directamente con su direccionamiento (por ejemplo M170.1).



## 7.2.2. Creación de los módulos

Después de analizar la ocupación de las variables, lo siguiente fue determinar, con ayuda de los GRAFCET, cuáles debían ser los bloques que se usarían para realizar el programa, ya fuesen nuevos o modificados del programa previo:

- **FC\_SalidasViales (FC20):** Puesto que el GRAFCET G0 representaba un proceso cíclico con unos pasos muy marcados, era conveniente el uso de una FC en la que se fuesen enlazando distintas condiciones para ir activando y desactivando estos pasos sucesivamente. En esta función también se determina el comportamiento de la instalación frente a fallos.
  - **DB\_PasosCicloViales (DB20):** Para la gestión de los pasos del ciclo de la FC20, se reservó un DB con un bit para cada uno de ellos, que solo sería igual a “1” cuando dicho paso estuviese activo.
- **FB\_PistonCiclo (FB20):** En cuanto al GRAFCET G1, al tratarse de un proceso igual en los tres cilindros, simplemente variando las condiciones para su activación, era un caso idóneo para el uso de un FB. En este, se especificaría el comportamiento de los pistones, además de las reacciones que debía tener cada uno en situación de fallo. Para un apoyo al mantenimiento de la planta, se incorporó un contador de ciclos del pistón, con tal de poder llevar un registro y hacer las revisiones pertinentes al llegar a un número establecido. La llamada del FB20 se realiza desde la FC20.
  - **DB\_Piston01 (DB21), DB\_Piston02 (DB22) y DB\_Piston03 (DB23):** Para el uso de los FB es imprescindible que tengan asociados un DB de instancia. En cada uno de ellos se guardarían los valores característicos de cada pistón (tiempo de salida, contador, etc.).
- **FC\_AlimentaciónMesa (FC11):** Un aspecto importante era el de adaptar el ciclo ya existente en el proceso con el nuevo. Tal y como se ha explicado, inicialmente la instalación ya era capaz de tener un control de los viales que se encontraban dentro de la mesa para ir avanzando la fila al llenarse, y dar el aviso al robot de que podía coger viales. Este tratamiento de datos de la entrada de viales a la mesa se recogía en la FC11. Para el correcto funcionamiento de la instalación, fue necesario sincronizar los pasos ya existentes con los nuevos, relacionados a los pistones. Además, se tuvo que depurar esta función ya que era un programa antiguo y contaba con redundancias, funciones en desuso y una falta de estructura, que se cambió para que siguiese la misma pauta que los nuevos bloques.
- **FC\_ConduccionCicloPiston (FC21):** Dado que el proceso contaba con la posibilidad de alternar el modo automático con el manual, se añadió también la FC21, que se encargaría de gestionar el cambio entre modos y la manera de actuar en manual, con especial hincapié a la seguridad del operario.
- **FC\_Alarmas (FC3):** Por otro lado, también tuvo que adaptarse el apartado encargado de gestionar las alarmas del proceso. El programa anterior ya contaba con un bloque dedicado a esto, la FC3. En cuanto a la seguridad, era imprescindible incluir condiciones de desactivación en caso de emergencia y actuaciones para asegurar la protección tanto de los operarios como del equipo.



- **DB\_HMI (DB29):** Por último, aunque esta parte queda explicada en el siguiente punto, se habilitó el DB29 para el tratamiento del HMI de la instalación, es decir, la pantalla con la que el operario puede interactuar con la máquina.

Todos los módulos incorporados a partir de este proyecto quedan expuestos en los anexos del documento, ya que representan una gran cantidad de páginas. El lenguaje en el que son mostrados es AWL ya que es el más compacto y comprensible a la hora de visualizar el código fuera del entorno STEP7, además de ser el lenguaje con el que se realizó la mayor parte del programa.

## 8. Supervisión, Control y Adquisición de Datos

El acrónimo inglés de Supervisory Control And Data Acquisition (Supervisión y Adquisición de Datos), SCADA, se emplea para nombrar a cualquier sistema (programa) que se encargue de vigilar el ciclo de un automatismo con el fin de optimizar su proceso. Con la gran evolución de la tecnología en el ámbito de la automatización, este campo ha cobrado gran importancia y mediante la realimentación, es decir, los ajustes constantes a partir de los datos obtenidos, se consigue una optimización que conlleva a una reducción considerable de los gastos. La principal función del SCADA es la de obtener datos (las variables del PLC, como por ejemplo las entradas mediante sensores) y a partir de estos datos graficar los valores, almacenarlos, activar alarmas si es necesario, etc.

Uno de los elementos más importantes del SCADA es el de la Interfaz Hombre-Máquina (HMI), que es el medio de comunicación entre la instalación y el usuario. Este aparato permite al operario tener un control de las distintas variables del proceso en tiempo real y da la posibilidad de actuar directamente sobre el ciclo de la máquina. Es habitual el uso de pantallas en cualquier tipo de máquina, que deben ser lo más sencillas e intuitivas posibles para que el usuario no tenga problema en manipularlas y evitar confusiones. Las funciones principales del HMI son: consultar el estado del proceso (por ejemplo comprobar si un motor está en marcha), conocer mediciones de campo (como podría ser obtener valores de la presión de una bomba), activar acciones (hacer salir un pistón, por ejemplo) e introducir valores y parámetros (como la velocidad de una cinta transportadora). Actualmente, las pantallas táctiles son muy comunes por su sencillez a la hora de utilizarlas, evitando tener que incorporar teclados u otro tipo de botones.

### 8.1. SCADA en Siemens

En cuanto a los HMI, como sucedía con los PLC, cada empresa cuenta con sus propios modelos que por más que sean parecidos entre las diferentes marcas, tienen características que los distinguen. Tal y como el PLC usado en este proyecto, el HMI debía ser de la marca Siemens. Dicha marca cuenta con distintas gamas según el uso que se le vaya a dar. En el caso de la estación de carga de viales del proyecto, se contaba con un aparato Simatic Comfort Panel, que ofrecen una gran funcionalidad y están pensados para aplicaciones de alto nivel. El modelo en concreto era el TP700 Comfort, con una pantalla táctil de 7 pulgadas.



**Figura 8.1:** HMI Siemens, modelo SIMATIC TP700 Comfort

Para la programación del HMI, la marca Siemens también cuenta con un programa propio. Originariamente, el entorno de diseño para los SCADA de Siemens era el WinCC (Windows Control Center – Centro de Control de Ventanas), cuya versión más básica es el WinCC Flexible. Pero con la evolución de la marca, se optó por crear un entorno que juntase las funciones tanto de programación (para evitar usar por separado el STEP7), como de diseño del SCADA (supliendo el uso del WinCC), por lo que nació el TIA Portal (Totally Integrated Automation Portal), que permitía programar todos los aspectos referentes a un PLC.

Dado que la instalación ya contaba con un HMI programado con el entorno TIA Portal v13, se decidió mantenerlo y programar la nueva versión de las pantallas con el mismo programa. Una ventaja de este programa es que permite trabajar sin ningún problema con proyectos realizados en STEP7, como era el caso. Simplemente era necesario identificar el PLC que se gestionaría y especificar las direcciones de las variables que querían tratarse.

Fue necesaria una breve formación para este entorno de diseño, aunque al tratarse de un programa muy intuitivo, esta fue rápida y se pudo hacer por cuenta propia.

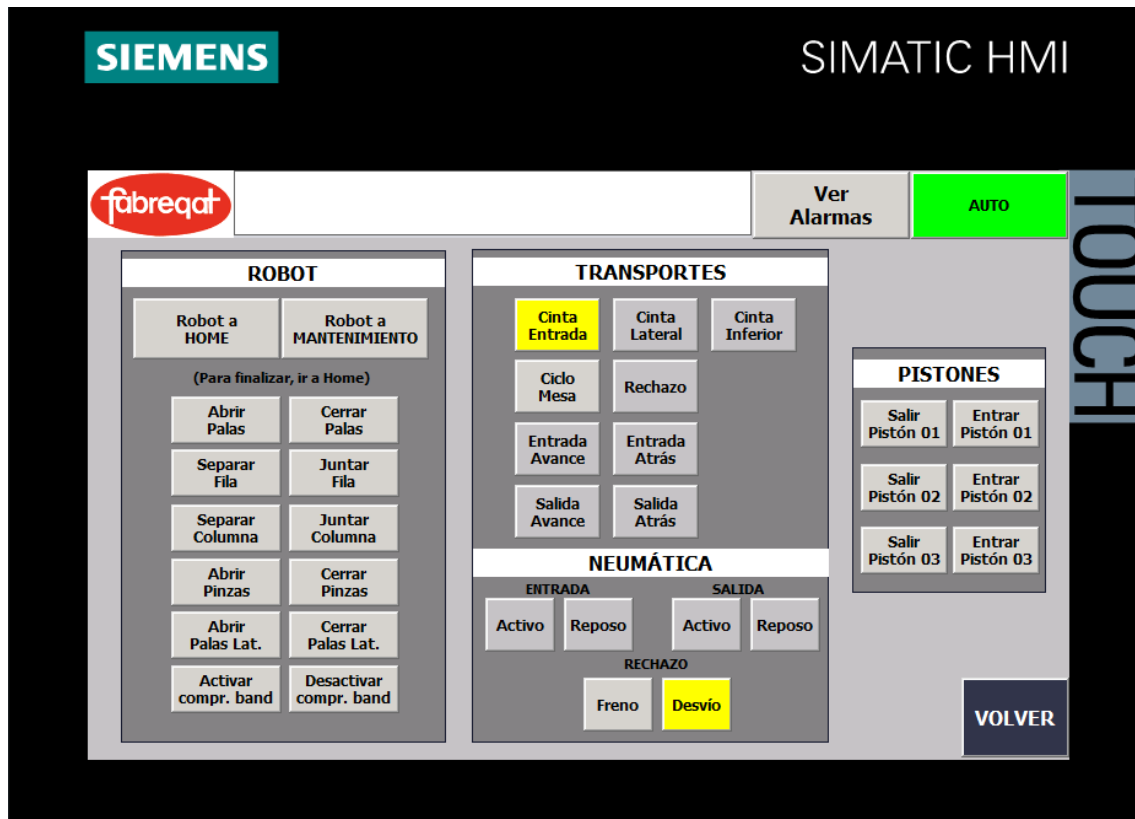
## 8.2. Interfaz Hombre – Máquina

El HMI previo contaba básicamente con tres pantallas importantes: la principal, que muestra un esquema de la instalación y facilita los datos de cuántos viales hay en la mesa, cuántos bloques se han llenado, etc.; la pantalla de estado, que indica en qué estado se encuentra el ciclo del brazo robótico para que los operarios puedan orientarse en caso de existir algún fallo; y la pantalla de manual, que permite activar manualmente distintas acciones de la instalación, como pueden ser los motores de las cintas de entrada de viales a la mesa.

La segunda de las pantallas que se han citado en el anterior párrafo no requería ningún cambio ya que el ciclo del brazo robótico no iba a alterarse (únicamente se desactivaría la separación de las pinzas a la hora de coger los viales de la mesa, pero esto se realizaba desde otro panel, el propio del robot), por lo que se mantendría la versión ya existente y por eso no se muestra en este proyecto.

Por otro lado, la pantalla del modo manual sí que era necesario cambiarla, ya que había que añadir un apartado que permitiese accionar manualmente la entrada y salida de los pistones (siempre y cuando se cumpliesen las condiciones de seguridad).

Tal y como puede verse en la *Figura 8.2*, se incorporó al HMI un bloque con el mismo diseño a la derecha de la pantalla que permite hacer salir o entrar los distintos pistones si está en modo manual. Para cambiar del modo automático al manual, es necesario accionar el botón verde “AUTO” que se encuentra en la esquina superior derecha de la pantalla, que cambiaría a amarilla con el nombre de “Manual” al entrar a este modo. Esta acción ya existía en el programa previo.



**Figura 8.2:** Pantalla de modo manual

Al accionarse la salida o entrada de cualquiera de los pistones, la condición activa siempre quedaría identificada ya que el botón cambiaría de color, pasando a amarillo, para que el operario sepa en todo momento cuál es la posición de cada pistón. Se puede apreciar que el entorno de diseño, además de tener la capacidad de poder forzar variables del programa del PLC, también permite animar la pantalla haciendo cambiar de color botones, provocando que parpadeen o incluso que desaparezcan con tal de hacer más visual la pantalla y facilitar el trabajo al usuario.

Por otro lado, al haberse añadido el ciclo de los pistones, resultaba interesante incorporar una pantalla en la que apareciesen todos los estados a modo de GRAFCET, diferenciando aquel que estuviese activo (cambiando el rectángulo a color verde). En la *Figura 8.3* puede verse la pantalla referente al ciclo de los pistones, que además cuenta con la representación de los contadores de ciclos de dichos cilindros, que cambiarían a color rojo en caso de superarse el límite establecido, y permitirían indicar al programa que han sido revisados o que se quiere reiniciar el conteo.

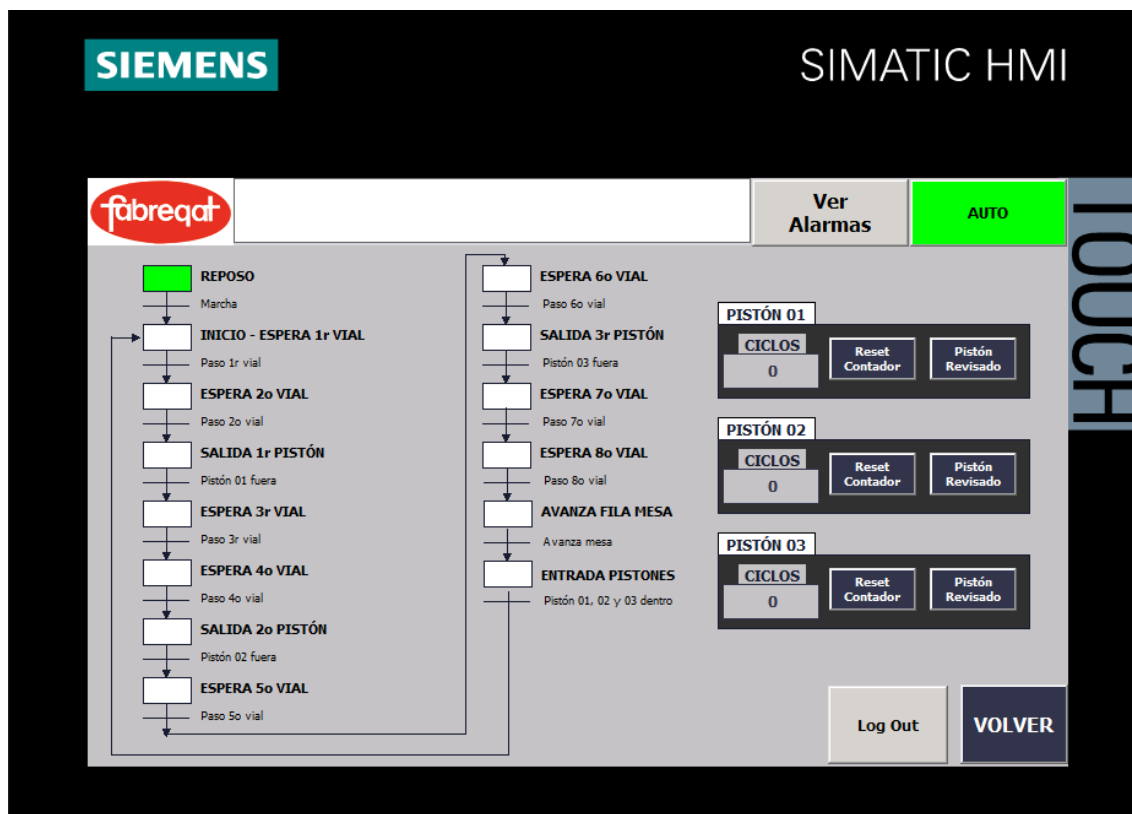


Figura 8.3: Pantalla del ciclo de los pistones

Por último, aunque el diseño de la pantalla principal ya era el correcto, al ser esta la que permitía acceder al resto de pantallas, tuvo que añadirse el direccionamiento, mediante el botón “Estado Pistones”, que llevase a la pantalla del ciclo de los pistones.

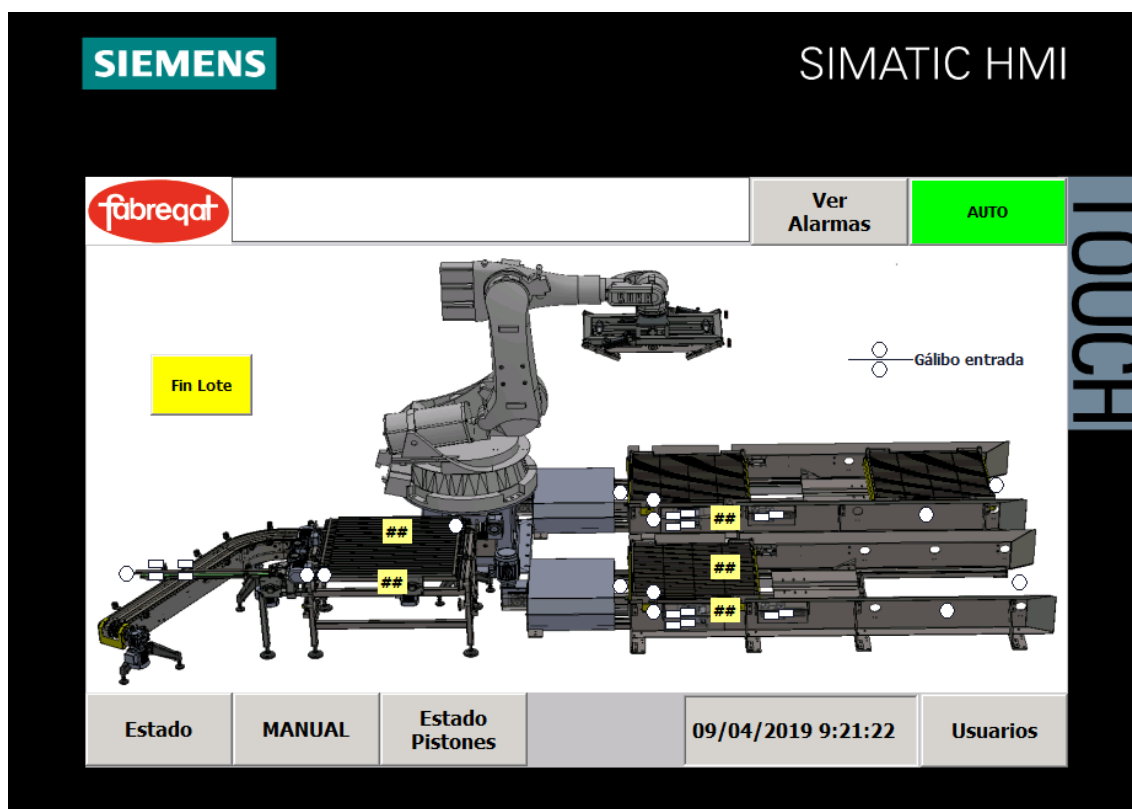
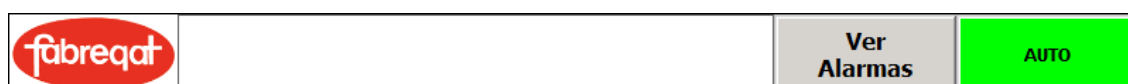


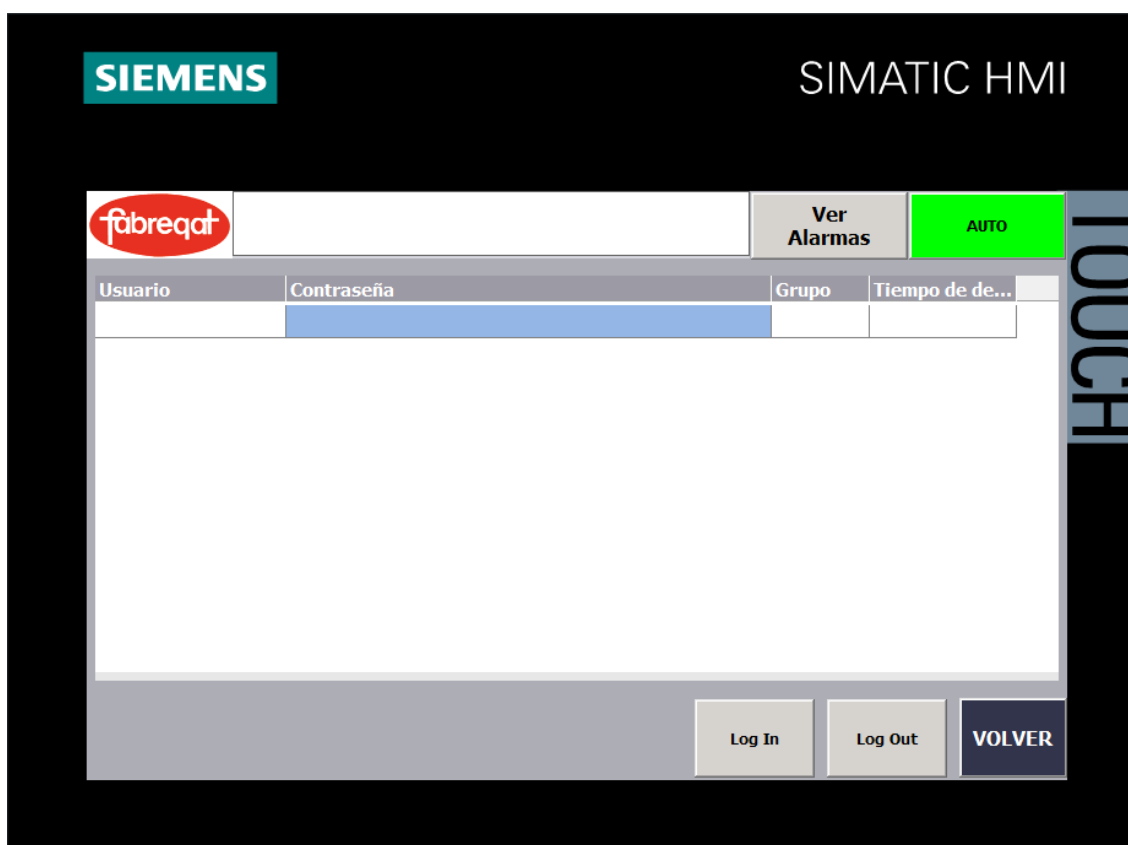
Figura 8.4: Pantalla principal del HMI

Además de los cambios que se han explicado en referencia a las distintas pantallas, todas ellas cuentan con un encabezado (parte superior) común, que no varía por más que se cambie la ventana. Este encabezado cuenta a la izquierda con el logotipo de la marca que realizó la instalación y en su parte central con un panel en blanco en donde aparecen las alarmas que hay activas para que, en caso de incidencia, pueda identificarse directamente el fallo sin necesidad de buscarlo el operario. Para que esto funcione, fue necesario añadir las nuevas alarmas que se habían creado con el programa de los pistones. Este encabezado, mostrado en la *Figura 8.5*, también permite ver un historial de alarmas mediante el botón “Ver Alarmas” y cuenta con la opción de cambiar de modo automático a manual y viceversa, como ya se ha explicado.



**Figura 8.5:** Encabezado HMI

Por último, otro punto importante en el HMI es el de la gestión de los usuarios. Por ejemplo, el modo manual únicamente pueden emplearlo los operarios de mantenimiento, por lo que es necesario identificarse mediante un usuario y una contraseña para tener acceso a este modo.



**Figura 8.6:** Pantalla de registro de usuarios

Después de realizar los cambios en el HMI de la instalación, por exigencias de la empresa ante cualquier cambio, se requirió la actualización del manual de pantallas que había proporcionado el fabricante, ya que en ciertos puntos estaba desfasado y además faltaban las partes referentes a las nuevas incorporaciones.

## 9. Diseño físico

A medida que fueron confeccionándose los programas del automatismo, por cuestiones de plazos y también para que los programas se adaptasen al diseño final, se combinó con la preparación del diseño físico de la implementación. En este apartado se pretenden justificar los elementos incluidos en el presupuesto detallado en el *Apartado 5.1.1* y complementarlo con el diseño final, aportando datos y medidas.

### 9.1. Electroválvulas

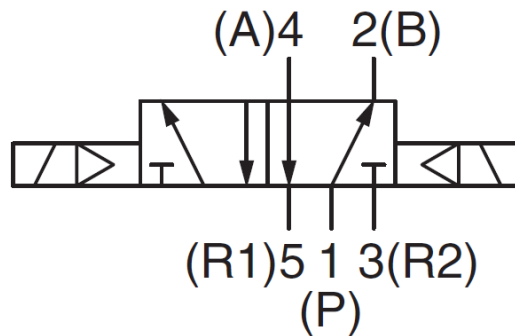
Con el fin de accionar los cilindros que se emplean como topes de las tiras de inyectables en la mesa de acumulación, la opción más sencilla era la de utilizar un sistema de válvulas que distribuyesen el circuito neumático hacia los cilindros. Estas válvulas son dispositivos diseñados para controlar el paso de un fluido, en este caso aire, a través de un conducto.

Los requerimientos esenciales para este dispositivo eran que los pistones pudiesen tener dos posiciones (fuera o dentro) y que estas se mantuviesen en el caso de que se cortase la alimentación, ya que una parada en la estación no podía provocar un movimiento de los pistones para retornar a su posición de equilibrio, tanto por motivos de seguridad como para asegurar la eficiencia del proceso. Por lo tanto, una condición indispensable era que las válvulas fuesen biestables, es decir, que pudiesen mantenerse en equilibrio en ambas posiciones. De este modo, si se daba un corte en la alimentación, permanecerían en la posición en la que se encontrasen. Por otro lado, la instalación no requería una presión excesivamente alta ya que la finalidad de los pistones no era aplicar ninguna fuerza, sino que simplemente la presión debía ser la suficiente para que el pistón saliese (o entrase) en un tiempo aceptable para no perjudicar el transcurso del ciclo, por lo que no eran necesarias válvulas de gama alta. Por último, las válvulas debían ser gobernadas mediante el PLC de la instalación, por lo que se usarían electroválvulas, que funcionan gracias al campo magnético de un solenoide, de ahí el nombre que reciben.

Para las instalaciones con cilindros biestables es común el uso de válvulas 5/2, es decir, con cinco vías y dos posiciones. Las dos posiciones son necesarias para los dos estados del pistón (dentro o fuera) mientras que las cinco vías, que son los orificios de salida o entrada de la válvula y se representan en la *Figura 9.1*, son empleadas para la entrada de aire (vía 1 de la figura), para el accionamiento de cada una de las posiciones (vías 2 y 4) y para el escape de aire (vías 3 y 5). Este tipo de accionamiento también podría realizarse con una válvula 4/2, en la que únicamente hubiese una vía de escape, común para ambas posiciones del pistón, siendo ligeramente más económicas pero con el inconveniente de la ralentización del cambio de estado. Es por esto que se optó por una válvula 5/2.

En la *Figura 9.1* puede apreciarse el esquema del tipo de válvula elegido. Las dos posiciones se representan con los dos rectángulos de la parte central, cada uno con cinco conexiones al exterior (las cinco vías). A los extremos de estos rectángulos, mediante otros rectángulos con un triángulo y una línea diagonal en su interior, se representa que ambas posiciones se accionan gracias a un electroimán servopilotado.

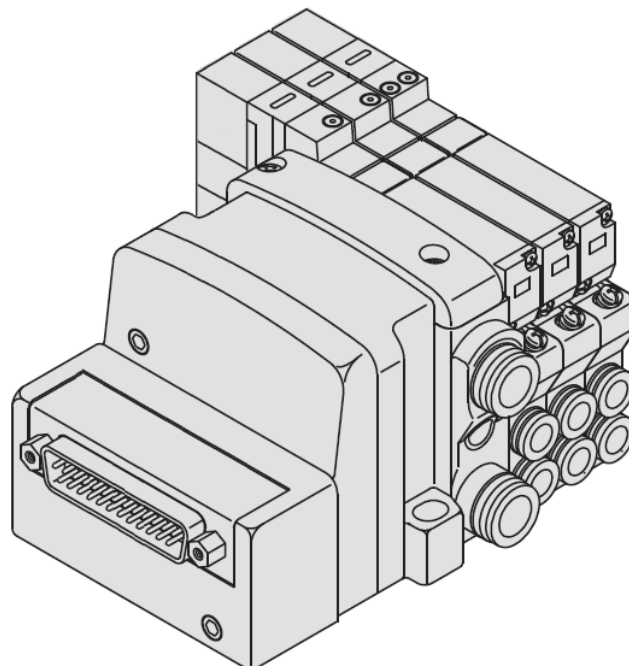




**Figura 9.1:** Esquema electroválvula 5/2

A partir de estas primeras especificaciones, se contactó con la multinacional japonesa SMC, puntera en la venta de productos para la automatización neumática. Entre las múltiples opciones de válvulas que ofrecían, había que buscar un modelo de electroválvulas pilotadas, dentro del grupo de 4 y 5 vías. De entre ellos se destacó el VQC, en especial el VQC2000, con un diseño compacto y de bajo consumo que se adaptaba perfectamente a los requerimientos de este proyecto. Al tratarse de tres cilindros los que se tenían que alimentar, se requería un bloque de tres estaciones (válvulas).

Una de las características importantes a tener en cuenta al elegir el bloque de válvulas era su modo de entrada eléctrica, es decir, el tipo de punto de conexión para la entrada de energía e información con tal de llevar a cabo el proceso. Debido a que lo que se buscaba era que el bloque fuese lo más compacto posible y que no requiriera tampoco un gran número de conexiones ni protección de ningún tipo (como podría ser protección al agua), se optó por un puerto de entrada mediante un multiconector sub-D de 25 pines, que iban directamente conectados interiormente a las salidas, sin necesidad de cables externos. Por otro lado, la alimentación sería de 24 V ya que era la utilizada en el resto de elementos de la instalación y la conexión sería mediante un cable de 6 mm de diámetro.

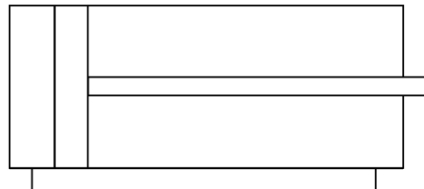


**Figura 9.2:** Bloque de 3 electroválvulas VQC2000 con multiconector sub-D

## 9.2. Cilindros compactos

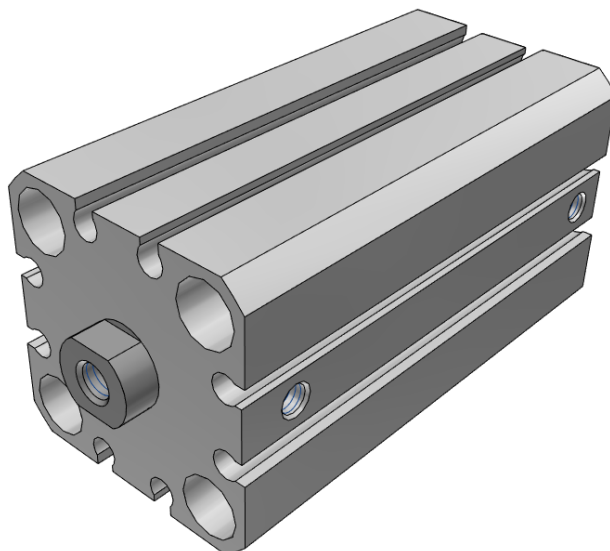
En cuanto a los topes para conseguir la separación de los viales en la mesa, tal y como se ha comentado, se iba a usar un juego de cilindros activados neumáticamente gracias a las electroválvulas explicadas en el anterior apartado.

Los cilindros neumáticos son un tipo de actuador, en este caso lineal, que se alimentan con aire comprimido con tal de hacer cambiar su posición a causa de la presión. Debido a la necesidad de que el sistema fuese biestable, era imprescindible el uso de un cilindro de doble efecto, es decir, con dos entradas de aire, una a cada extremo del cilindro, para que cada una de las posiciones del vástago que ejercería de tope (dentro o fuera) se activase mediante la entrada de aire comprimido por uno de los dos compartimientos, mientras que el otro se vaciaba. Cada una de estas entradas de aire son las controladas por las dos salidas de la electroválvula. El problema de haberse usado un cilindro de efecto simple sería que de cortarse la alimentación, estos solo tendrían una posición de equilibrio ya que el retorno a la posición inicial se ejecuta mediante un muelle.



**Figura 9.3:** Esquema cilindro de doble efecto con vástago

Del mismo modo que con los otros elementos de la instalación, se contactó con la empresa proveedora, en este caso también SMC para decidir cuál de los productos que ofrecían se adecuaba a los requerimientos del proyecto. No era necesario un cilindro que ejerciese fuerza, por lo que la condición más importante era que ocupase el mínimo de espacio. Por este motivo y porque la forma cuadrada resultaba más cómoda para acoplar el cilindro a la estructura, se optó por los cilindros compactos. De entre los modelos que ofrecía la empresa, se escogió la serie CQS ya que se adaptaba al presupuesto y cumplía con las necesidades.



**Figura 9.4:** Cilindro neumático CDQS B25-45DC

Dentro de este modelo, se eligió un cilindro que pudiese incorporar sensor magnético para identificar su posición. Las medidas del cilindro fueron de 25 mm de diámetro, con un recorrido de 45 mm ya que simplemente debía moverse para tapar o dejar libre el paso por la primera fila de la mesa.

Por otro lado, se incorporó al extremo del vástago un recubrimiento de Polietere tercetona (PEEK), un polímero con bajo coeficiente de fricción para evitar dañar los viales, y una alta resistencia al uso, desgaste y abrasión. Para este caso se determinó un diámetro de 12 mm, que coincidiese con las barras que ejercerían de límites para los carriles de la mesa. Además, para facilitar el ciclo, la terminación de este extremo de PEEK era en punta para poder entrar dentro de la barra anteriormente citada, ya que se trataba de un tubo hueco.

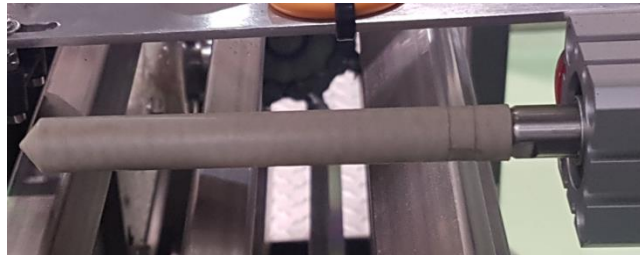


Figura 9.5: Recubrimiento del vástago de PEEK

### 9.2.1. Detectores magnéticos

Era necesario tener un control en todo momento de la posición de cada uno de los cilindros. La empresa SMC ofrece múltiples detectores magnéticos de posición para los actuadores neumáticos. Por motivos de optimización de espacio, la mejor opción era la de sensores montados directamente en las ranuras del cilindro.

Los detectores magnéticos son un tipo de sensores capaces de revelar la proximidad de un objeto gracias a la alteración del campo magnético. Dentro de estos detectores magnéticos hay dos alternativas básicas:

- **Detectores de estado sólido:** Funcionan gracias al efecto Hall, es decir, cuentan con un material conductor por el que circula corriente (circuito azul de la Figura 9.6). Al acercarse un imán, a causa de la fuerza de Lorentz, las cargas negativas del conductor se desvían a un extremo de este, quedando polarizado. Se obtiene una tensión (voltaje de Hall, circuito rojo) entre sus extremos, perpendicular a la que se tenía sin el imán.

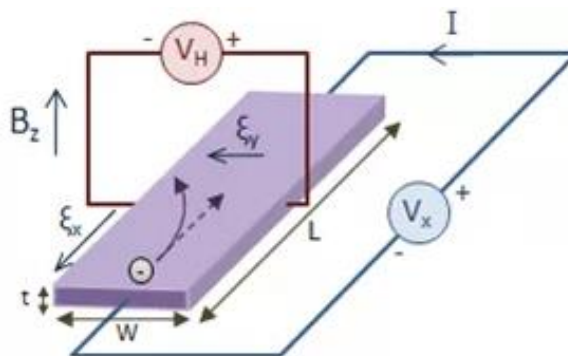
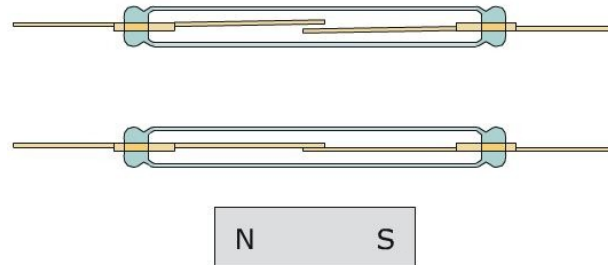


Figura 9.6: Esquema efecto Hall

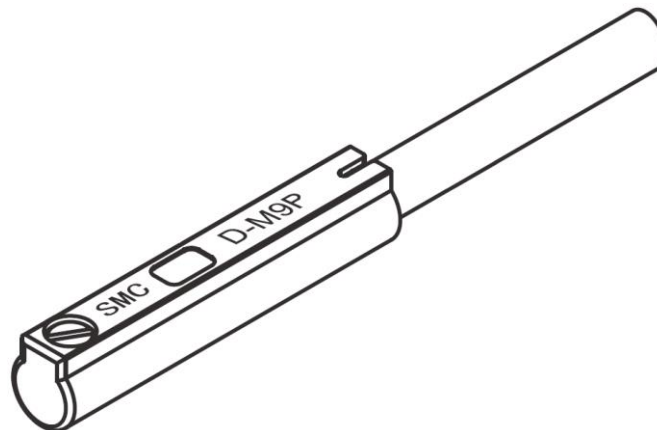
- **Detectores de tipo *reed switch* o de lengüeta:** Cuentan con un par de contactos cerrados al vacío en el interior de un tubo de vidrio. En condiciones normales, los contactos se encuentran separados pero al acercarse un imán, se unen para cerrar el circuito.



**Figura 9.7:** Esquema contacto reed switch

Los detectores *reed switch* son más económicos que los de estado sólido pero tienen una menor durabilidad, ya que pueden acabar polarizándose o rompiéndose debido a que cuentan con una parte móvil, al contrario que los de estado sólido. La diferencia de precio no era muy significativa y resultaba interesante que los detectores fuesen resistentes, por lo que se acabó optando por unos detectores de estado sólido.

El modelo elegido fue el D-M9P, con una conexión PNP de tres hilos (dos para la alimentación y uno para la salida del sensor), con un conector M8 de 3 pines. Estos sensores se montan directamente en las ranuras de los cilindros. El cilindro tiene internamente un indicador de posición magnético, que activa el detector tras su paso por la posición en la que esté montado.



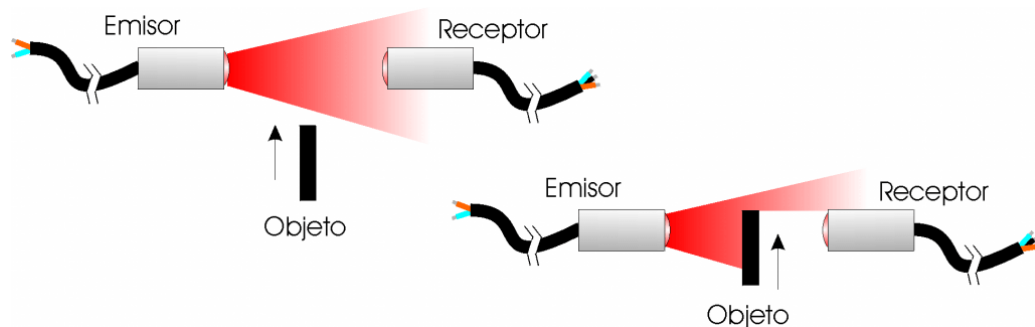
**Figura 9.8:** Detector magnético de posición D-M9P

### 9.3. Fotocélulas

Con tal de tener un control de la posición de los viales dentro de la mesa, resultaba interesante incorporar sensores que fuesen capaces de detectar los inyectables. Se optó por poner un sensor al lado de cada cilindro para que el sistema pudiera tener constancia de que las tiras estaban bien posicionadas y el pistón podía salir. Esto se podría haber realizado mediante el uso de temporizadores, evitando así los sensores, pero debido a que se buscaba reducir al mínimo los fallos de la instalación, se corría el riesgo que por culpa de algún encallamiento de las botellas, los pistones pudiesen salir sin estas haber llegado, por lo que el gasto en los sensores era justificable.

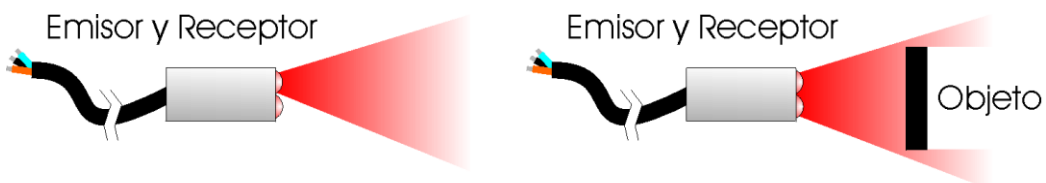
La opción más sencilla para detectar las tiras de inyectables era incorporando sensores fotoeléctricos de posición. En la empresa era habitual el uso de fotocélulas, por lo que la búsqueda fue encaminada a este tipo de dispositivos. Estos aparatos emiten un haz de luz (visible o infrarrojo) desde un elemento emisor de luz. Este haz de luz ha de llegar a un elemento receptor que dispone el detector ya que, de lo contrario, significa que existe un objeto que lo está interceptando, activando la salida del sensor de posición. Existen tres grandes tipos de sensores fotoeléctricos:

- **Fotocélulas de barrera:** El emisor y el receptor están instalados en dos dispositivos distintos, que han de colocarse alineados para quedar encarados. En el caso de que aparezca un objeto entre estos dos elementos, el haz de luz no podrá llegar con normalidad al receptor. Son los dispositivos más fiables y con un mayor alcance.



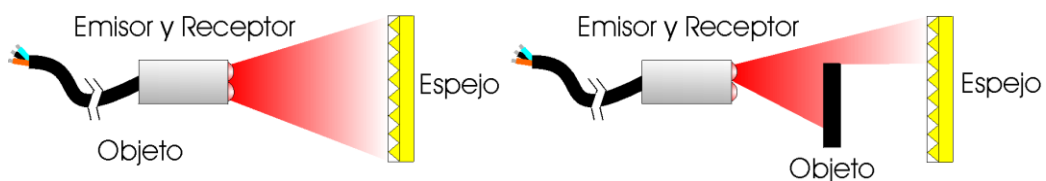
**Figuras 9.9a y 9.9b:** *Fotocélula de barrera (desactivada/activada)*

- **Fotocélulas autorreflexivas:** El emisor y receptor se encuentran en el mismo dispositivo. Si se sitúa un objeto en la trayectoria del haz de luz, este es reflejado por el cuerpo y el receptor capta la luz reflejada. Son el tipo más económico de fotocélulas ya que su precisión es menor, aunque resultan indispensables en ocasiones en las que el espacio es reducido.



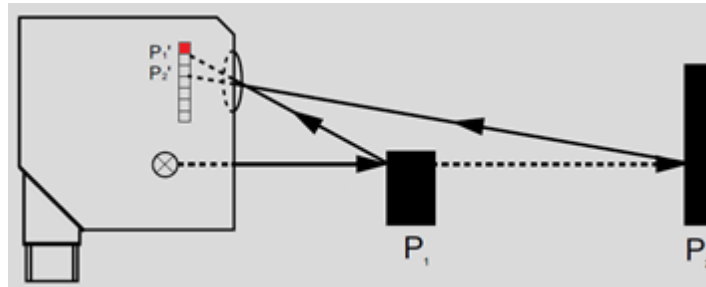
**Figuras 9.10a y 9.10b:** *Fotocélula autorreflexiva (desactivada/activada)*

- **Fotocélulas reflexivas con reflector:** Del mismo modo que las fotocélulas autorreflexivas, cuentan con emisor y receptor en un mismo dispositivo. Para el funcionamiento de este tipo de elementos, es necesario incorporar un reflector (espejo) que devuelva el haz de luz en caso de no existir ningún objeto que la intercepte. Consiguen detectar cuerpos a distancias mayores que las fotocélulas autorreflexivas pero no llegan a la precisión de las fotocélulas de barrera.



**Figuras 9.11a y 9.11b:** *Fotocélula reflexiva con reflector (desactivada/activada)*

Las fotocélulas se iban a colocar en la misma estructura que los cilindros, frente a la primera fila de la mesa. Estas debían ser capaces de detectar la existencia de inyectables pero, en su ausencia, no podían confundir la lectura con los viales de la segunda fila de la mesa. Esta condición, sumada a la imposibilidad de incorporar un reflector en la instalación, exigía que las fotocélulas fuesen autorreflexivas y además que contasen con supresión de fondo. Esta característica es propia de algunas fotocélulas autorreflexivas y permite calcular la distancia del objeto a detectar. Gracias a esto, es capaz de ignorar la existencia de otros objetos que pudiesen reflejar la luz, pero que se encuentran a una distancia mayor de la esperada, como serían en este caso las tiras de la segunda fila. Para ello utilizan los principios de la triangulación.



**Figura 9.12:** Principio triangulación (suspensión de fondo)

Con estas especificaciones se optó por realizar un estudio de las fotocélulas con las que se contaba en la empresa y comprobar su funcionamiento experimentando con tiras de inyectables. Uno de los hándicaps de este producto es que son transparentes, hecho que dificulta la reflexión de la luz. Esto, sumado a que este tipo de fotocélulas son las menos precisas, provocaba que fuese necesario un modelo de alta precisión.

La empresa disponía de diversos modelos de fotocélulas de la multinacional alemana IFM, dedicada a la producción de sensores. El objetivo era encontrar alguno de ellos que se adaptase a las especificaciones, es decir, que fuesen autorreflexivas con supresión de fondo, que el alcance fuese compatible y que pudiesen detectar objetos transparentes. De entre todos los modelos, destacó el OJ5158, que es un tipo de fotocélula con sensor láser, especializado en la detección de objetos pequeños, por lo que era más precisa que el resto. Durante las pruebas realizadas no dio ningún problema con la detección de las botellas transparentes a la distancia a la que estarían en la instalación. Además, el precio era aceptable, por lo que se escogió este modelo para la instalación.



**Figura 9.13:** Fotocélula OJ5158

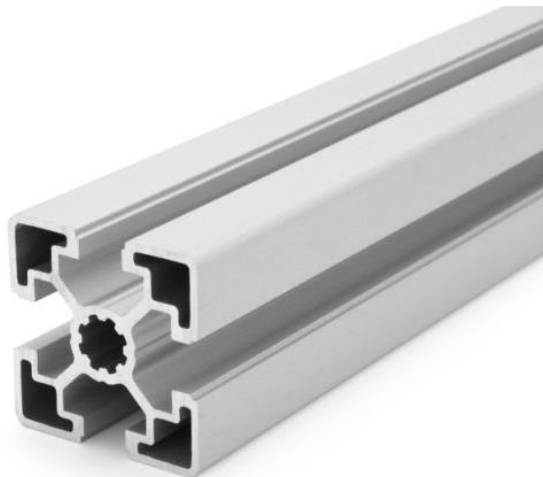
Para este tipo de sensores se utiliza un conector M8 con 4 pines.



## 9.4. Estructura

A causa del espacio reducido con el que se contaba en la instalación, el montaje no podía ser de grandes dimensiones. Además, debido a la gran precisión posicional que requería, resultaba interesante que pudiera ajustarse una vez instalado. Por las características previamente expuestas, se decidió que la mejor opción era diseñar una estructura formada por perfiles de aluminio ranurados, que permitiesen el ajuste tanto de altura como de posición de los cilindros, así como de los carriles de la mesa y de las fotocélulas. La ventaja de tratarse de aluminio era que no representaría una instalación pesada.

Por las medidas de los otros elementos y de la mesa, se escogió un perfil de 35x35 mm. Estos perfiles tienen la ventaja de que pueden acoplarse entre sí gracias a unos tornillos con forma de T que encajan con los raíles, pudiendo bloquearse su salida. Por otro lado, también pueden incorporarse soportes para las fotocélulas o pistones mediante el mismo método.



**Figura 9.14:** Perfil ranurado de aluminio

Con dos grupos de tres piezas de perfil de aluminio se podía construir la estructura principal, formando dos arcos, uno para cada frente de la mesa. Estos arcos deberían ir anclados al suelo ya que serían el soporte de la instalación.

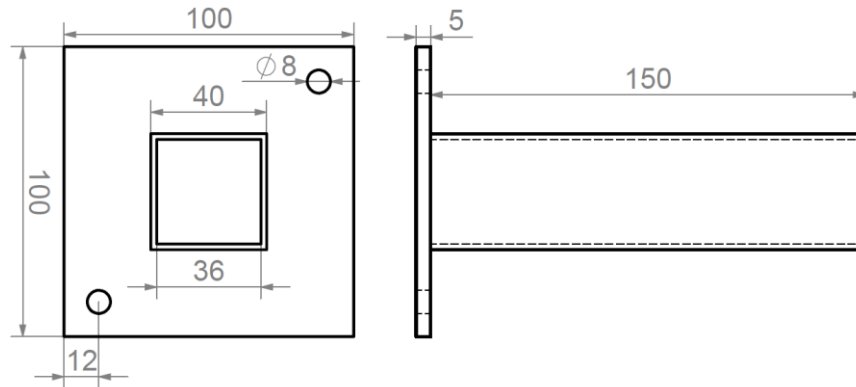
A partir de esto, lo siguiente fue buscar la manera de formar cuatro carriles en la mesa que forzasen la disposición de los inyectables del mismo modo que en las bandejas. Se optó por incorporar tres barras cilíndricas perforadas de aluminio, con el mismo diámetro que el extremo de los pistones. Estos carriles debían empezar a partir de la segunda fila de la mesa, por lo que requerían de soportes que los mantuviesen a cierta altura de la mesa para evitar el roce con ella.



**Figura 9.15:** Tubo de aluminio

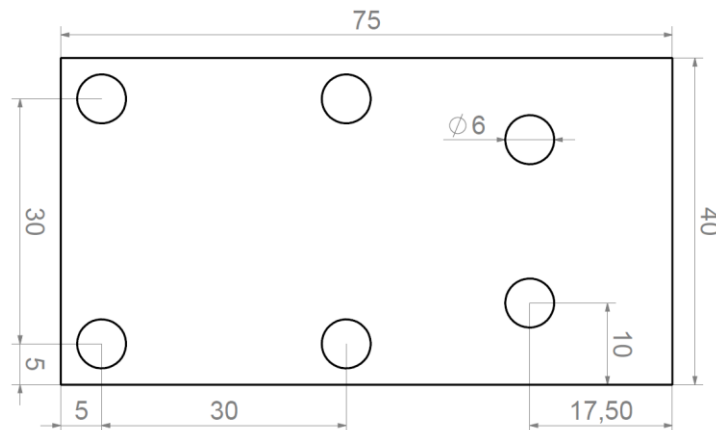
Por último, con tal de sujetar cada uno de los elementos, se diseñaron distintos soportes para su mecanización:

- **Estructura:** Con el fin de anclar al suelo los dos arcos que servirían como estructura básica de la instalación, hicieron falta cuatro pies que se adaptasen a los perfiles de aluminio. Estas piezas se solicitaron de acero.



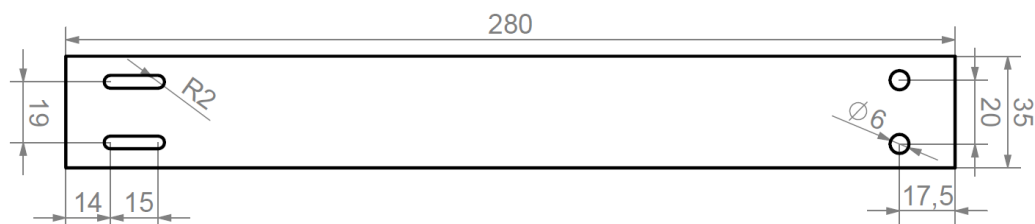
**Figura 9.16:** Soportes para la estructura de aluminio

- **Cilindros:** Para la sujeción de los pistones, se mecanizaron tres pletinas de aluminio de 5 mm de grosor que permitiesen anclar los cilindros mediante cuatro tornillos. También se habilitaron dos orificios para la unión a la estructura del perfil de aluminio.



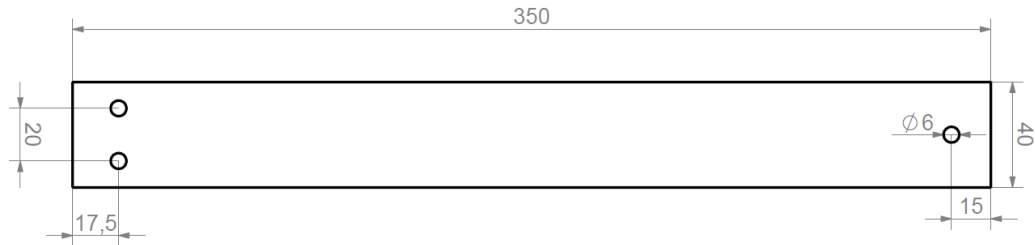
**Figura 9.17:** Pletinas para la sujeción de los pistones

- **Fotocélulas:** Con tal de disponer las fotocélulas a la menor distancia posible de los viales de la mesa, se diseñaron tres pletinas de aluminio de 3 mm de grosor. Por un lado, contaban con dos agujeros para su unión con el perfil de aluminio y por el otro contaban con dos agujeros colisos que permitirían fijar las fotocélulas con la posibilidad de variar su distancia a la mesa.



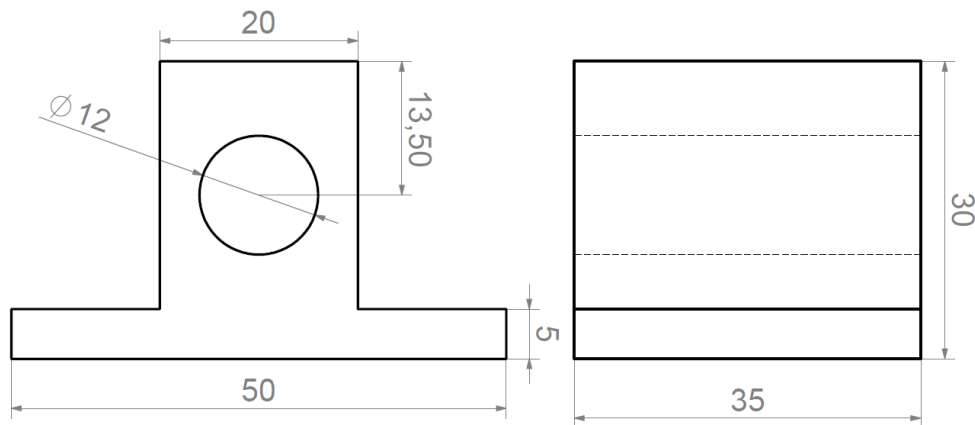
**Figura 9.18:** Pletinas para la sujeción de las fotocélulas

- **Carriles:** Por último, las barras para los carriles debían estar sujetas a ambos lados de la mesa. Estos soportes no podían provocar colisiones con el robot y además tenían que dejar libre la primera fila de la mesa para que pasaran las tiras de inyectables. Es por esto que se decidieron mecanizar tres pletinas de aluminio de 10 mm de grosor para sujetar las varillas a la estructura por el lado de los pistones.



**Figura 9.19:** Pletinas para la sujeción de las varillas para los carriles

Para el otro lado de la mesa se fabricaron tres piezas de aluminio como las mostradas en la *Figura 9.20*, con un agujero pasante con tal de poder regular la posición de los carriles.



**Figura 9.20:** Piezas para la sujeción y ajuste de las varillas para los carriles

Cabe añadir que se adquirieron canaletas de policloruro de vinilo (PVC) para poder pasar los cables y los tubos de la instalación neumática por la estructura y que estuviesen protegidos. Estas iban también unidas al perfil de aluminio.



**Figura 9.21:** Canaleta para cables y tubos

## 10. Implementación

### 10.1. Instalación eléctrica

Llegado el momento de implementarse físicamente este proyecto, lo primero que pudo realizarse fue la instalación eléctrica, ya que esta no requería que se parase la marcha de la carga de viales. Debido a que ya se contaba con un PLC, únicamente era necesario conectar las nuevas entradas y salidas que requeriría la automatización.

Tal y como se había comprobado en el *Apartado 7.2.1*, el PLC ya contaba con suficientes módulos de entradas y salidas digitales. Estos ya estaban alimentados por el propio controlador, por lo que simplemente se tenían que habilitar los conectores adecuados para cada tipo de sensor o actuador, conectándolos a estos módulos. Para ello, el cuadro eléctrico contaba con un grupo de borneras, que sería el punto de empalme.



Figura 10.1: PLC utilizado para el proyecto

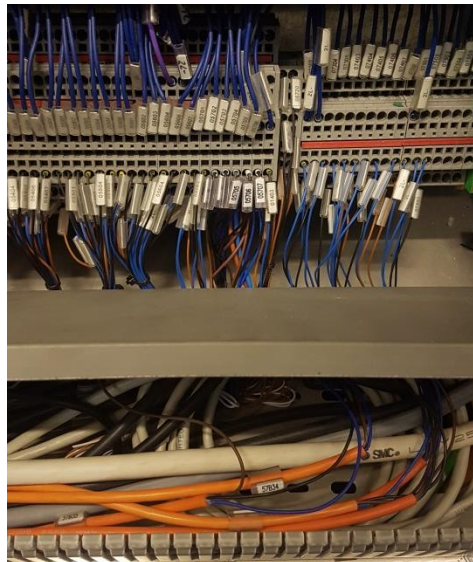


Figura 10.2: Borneras para la conexión de las entradas

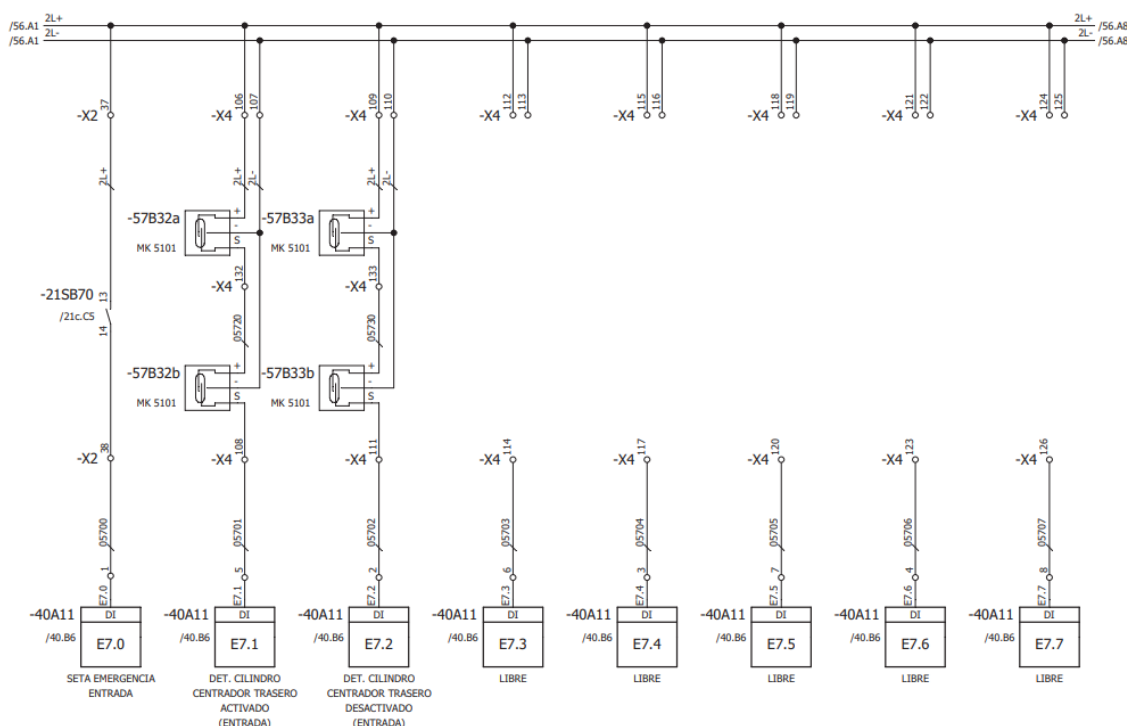
En la *Figura 10.1* pueden apreciarse los módulos de entradas y salidas digitales (a la derecha). Por otro lado, en la *Figura 10.2* puede verse la conexión de cables para las entradas, uniéndose en la bornera roja los conectores (naranjas) que irían hacia la instalación con tal de conectarse a los sensores. Estos conectores, tal y como se había comentado en el *Apartado 9.2.1* y el 9.3, serían M8 de 3 y 4 pines para los detectores magnéticos y las fotocélulas respectivamente.



**Figura 10.3:** Conectores M8 de 3 y 4 pines

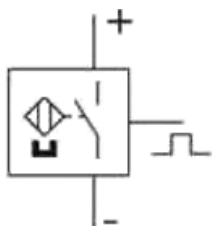
Por parte de las salidas, es decir, las electroválvulas para los cilindros neumáticos, el conector debía ser un multiconector sub-D de 25 pines tal y como se había justificado en el *Apartado 9.1*.

En cuanto a los esquemas eléctricos, estos debían ser actualizados con las nuevas entradas y salidas. El problema era que la empresa no contaba con el entorno de programación requerido, por lo que esta acción la realizaría la empresa de instalaciones. Estos no estuvieron disponibles antes de la presentación de este proyecto, por lo que se presenta a modo explicativo el esquema de las entradas previo a la actualización.

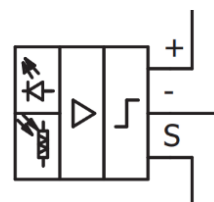


**Figura 10.4:** Esquema eléctrico byte 7 de entradas

En las entradas actualizadas habría que añadir el tipo de sensor (fotocélula, sensor magnético, etc.) y el código del cable, para facilitar su identificación.



**Figura 10.5a:** Símbolo detector magnético



**Figura 10.5b:** Símbolo fotocélula



## 10.2. Instalación neumática

Para lograr que funcionasen los cilindros neumáticos, se requería que la instalación estuviese alimentada neumáticamente. Debido a que en la estación de carga ya se utilizaban elementos neumáticos, dicha instalación ya existía y simplemente se tuvo que conectar la entrada de aire a las electroválvulas, y estas a los cilindros.



**Figura 10.6a:** Alimentación circuito neumático



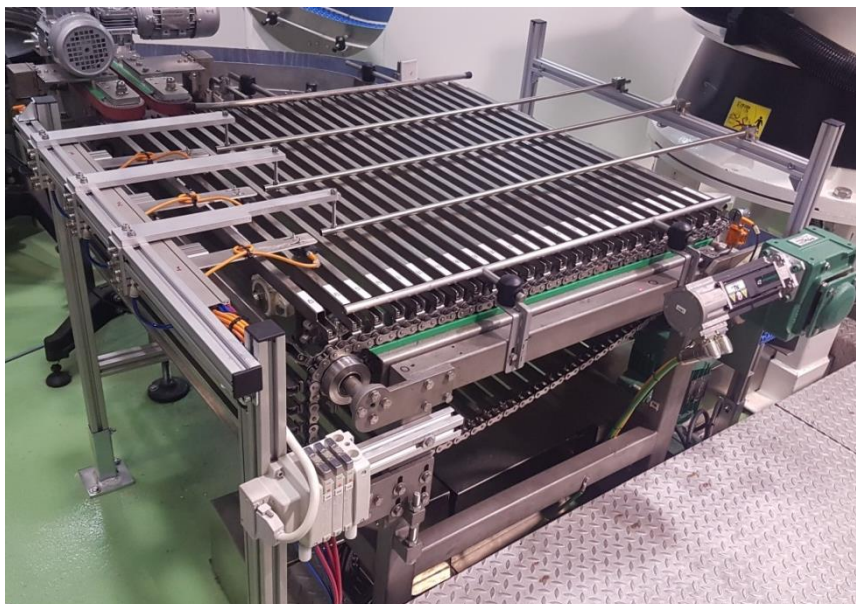
**Figura 10.6b:** Grupo electroválvulas

## 10.3. Instalación mecánica

Para el montaje de la estructura sí que era necesario que la estación estuviese parada. En primer lugar se instaló la base formada por los perfiles de aluminio, para poder ir incorporando los demás elementos que irían sujetos a estos.

Uno de los puntos más importantes era asegurar que el brazo robótico no colisionase con la estructura a la hora de ir a coger los viales. Tras la instalación, pudo comprobarse que parte de uno de los soportes para los carriles interrumpía la trayectoria del brazo, por lo que tuvo que cortarse para evitar que esto sucediese. Este corte no afectaba al montaje.

El resultado de la estructura una vez instalada se muestra en la *Figura 10.7*. En ella se puede ver la base formada con los perfiles de aluminio, los carriles para la mesa y los correspondientes soportes.



**Figura 10.7:** Mesa de acumulación tras la implementación trabajada en este proyecto



En la *Figura 10.8* puede apreciarse un conjunto de cilindro neumático y fotocélula.



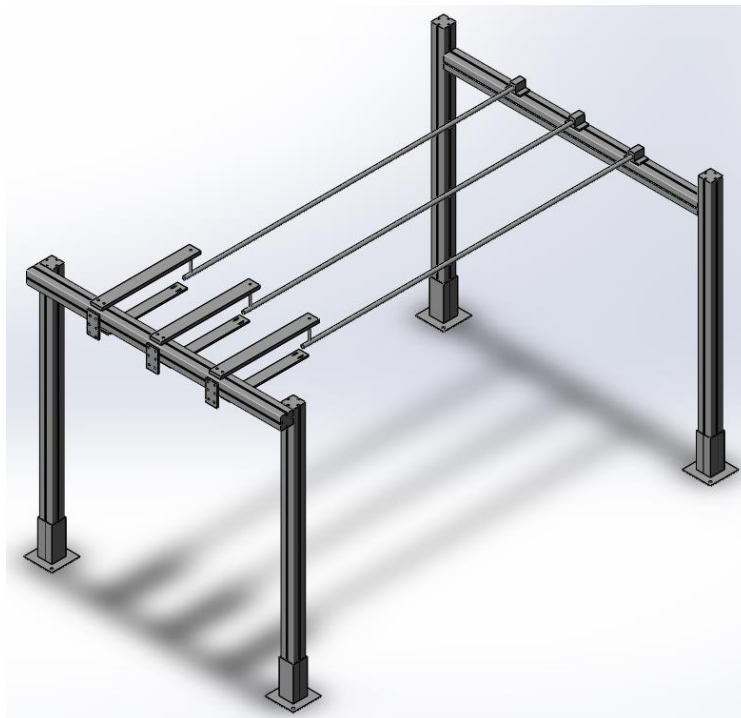
**Figura 10.8:** *Conjunto cilindro neumático – fotocélula*

Una vez montada la instalación, había que ajustar la posición de los carriles y pistones para asegurar su máxima eficiencia. Por otro lado, también era necesario calibrar los distintos sensores: las fotocélulas se ajustaban directamente del dispositivo, indicando las situaciones de lectura y ausencia de objeto con tal de poder emplear la supresión de fondo; los detectores magnéticos habían de posicionarse en el punto deseado para que realizasen la medición. En la *Figura 10.9* se muestra una imagen en detalle del montaje durante el proceso de calibración, con la fotocélula activada (luz naranja) y de ajuste de los carriles y pistones, con un cilindro salido y con los viales en la mesa.



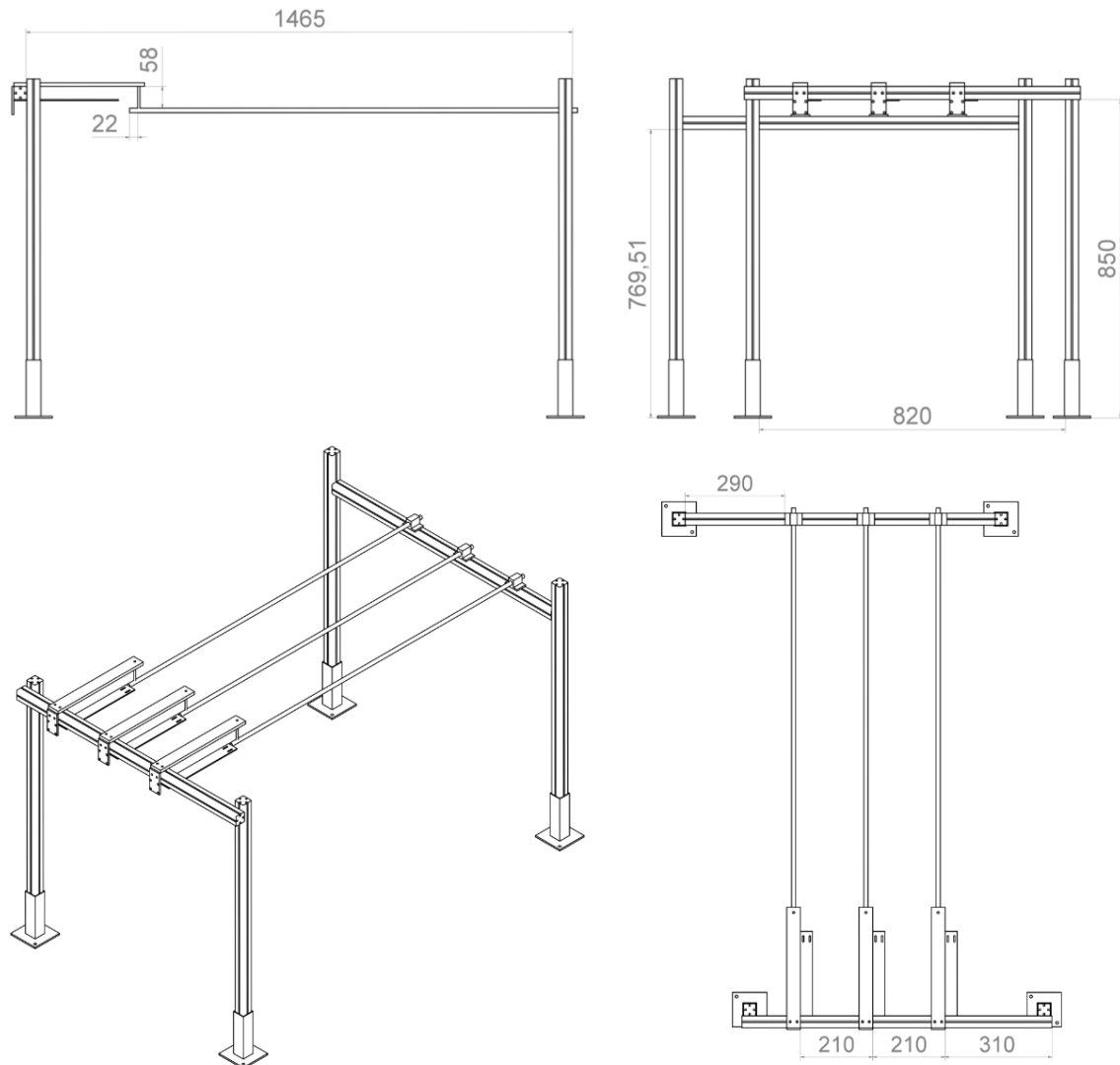
**Figura 10.9:** *Foto en detalle durante la calibración*

En la *Figura 10.10* se muestra el diseño en 3D del montaje mediante *SolidWorks*.



**Figura 10.10:** *Diseño en 3D del montaje*

En la *Figura 10.11* se muestran las medidas definitivas de la instalación más representativas.



**Figura 10.11:** Planos instalación (orientativos)

## 10.4. Carga del programa

En cuanto la instalación estuvo implementada, había que transferir del ordenador al PLC el programa para realizar las primeras pruebas.

Para la conexión entre el PC donde se encontraba el programa del automatismo en STEP7 y la CPU del PLC, se contaba básicamente con las siguientes opciones:

- **Profibus:** Es el bus de campo más empleado por las grandes marcas de automatización. Mediante la conexión DP, es posible un intercambio de datos cíclico y rápido. Aunque esté pensado principalmente para conectar distintas instalaciones a lo largo de la planta, también puede utilizarse para conectar con el nivel de control central (en este caso, el PC). El Profibus DP tiene una velocidad máxima de transmisión de 12 Mbit/s.

En la *Figura 10.12* puede apreciarse el conector de Profibus DP, que es el mismo que tenía la conexión MPI, más antigua a este.



**Figura 10.12:** Conector Profibus DP

- **Profinet:** Emplea la tecnología de Industrial Ethernet y es la opción más rápida para la transmisión de grandes cantidades de datos. Puede llegar a una velocidad de 100 Mbit/s. Generalmente se emplea para conexiones con el nivel de control. Su conexión es nombrada PN.



**Figura 10.13:** Conector PN

- **Wifi:** Ya que la CPU puede estar conectada a la red informática, existe la posibilidad de conectarse a distancia mediante conexión Wifi. Es la opción más lenta para la transmisión de datos.

En este caso, debido a que el ordenador ya cuenta con entrada Ethernet, la opción más sencilla para conectarse a la CPU era mediante Profinet. Si hubiera querido usar Profibus, hubiese sido necesario un adaptador de PN a DP para establecer conexión con el ordenador, por lo que no generaba ninguna ventaja.

En cuanto a la pantalla del HMI, las opciones de conexión para la carga del programa eran iguales, por lo que se empleó el mismo método.

## 11. Puesta en marcha

### 11.1 Adaptación con el ciclo previo

Debido a que la instalación ya contaba con un ciclo en funcionamiento, los principales problemas que surgieron durante las primeras pruebas con el nuevo programa fueron por incoherencias entre los dos ciclos: el de conteo de tiras de inyectables que entraban a la mesa (ya se realizaba este control de los viales que estaban en la mesa para poder avanzar la fila cuando estuviese llena), y el de las salidas de los pistones.

Una opción que ofrece el entorno de programación STEP7 es la de incorporar tablas de variables en las que pueden añadirse entradas, salidas, marcas, direcciones de DB o incluso contadores. La finalidad de estas tablas es la de poder consultar el estado de las variables una vez se está conectado al PLC en marcha, pudiendo llegar a forzar sus valores. Este aspecto resulta muy útil en la puesta en marcha de la instalación, sobre todo para detectar errores y comportamientos inesperados.

## 11.2. Parada de la cinta de entrada de viales

Las primeras pruebas para la puesta en marcha se realizaban alimentando la mesa con viales manualmente, con una cadencia menor a la que se tendría en producción normal. Es por ello que una vez que el ciclo funcionó en estas condiciones, había que asegurar su correcto comportamiento en una situación de producción habitual.

Uno de los problemas que podían aparecer era que llegase a la mesa un vial mientras el pistón estuviese saliendo. En el ciclo previo, cuando una tira llegaba a la mesa mientras se estaba avanzando una fila, las cintas de entrada de viales se paraban hasta que el movimiento terminaba. El objetivo era actuar del mismo modo con el caso inicialmente expuesto. Se había comprobado que con la frecuencia de salida de inyectables del envasado, era posible realizar estas pausas sin afectar a la producción.

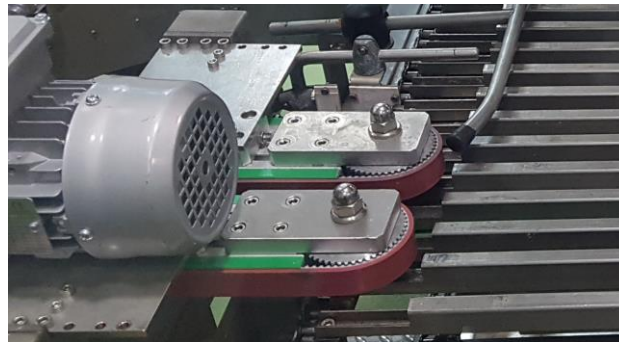


Figura 11.1: Cintas de entrada de viales a la mesa

Volviendo a la Guía GEMMA, se podía incluir en el módulo de modos de marcha. Considerándolo como una situación de fallo (llegada de un vial mientras un pistón está en movimiento), se requería la parada de la cinta de entrada pero el ciclo continuaba con normalidad. Es por ello que se emplea simplemente el modo D3, además del F1.

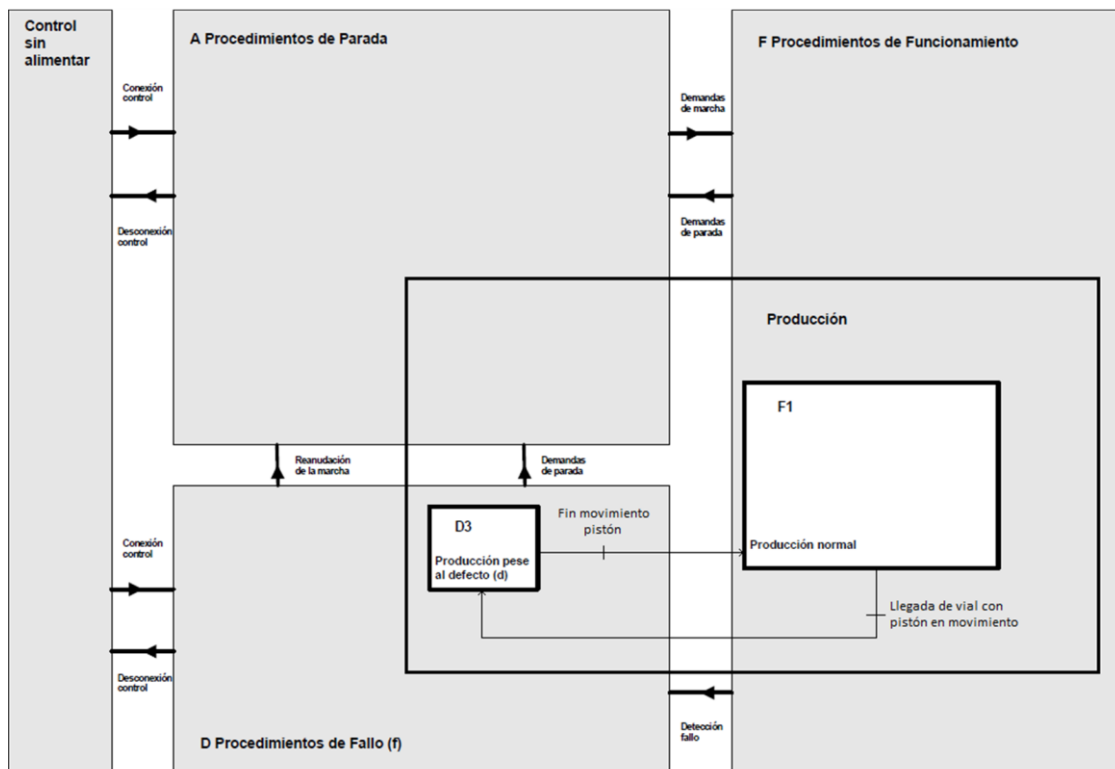
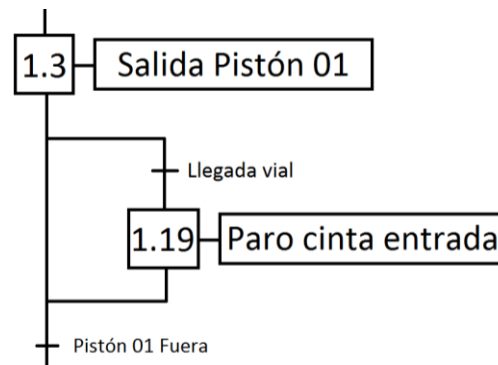


Figura 11.2: Diagrama Guía GEMMA para el control de la cinta de entrada

Estas condiciones se tenían que añadir a la primera versión del programa y modificarían el GRAFCET de producción G1. En la *Figura 11.3* se muestra cómo quedaría modificada una parte de este diagrama, siendo aplicable el cambio al resto de pistones.



**Figura 11.3:** Segmento del GRAFCET G1 con el paro de la cinta de entrada

### 11.3. Comprobación alarmas

Una vez el ciclo estaba listo para una producción normal, había que comprobar que las alarmas incorporadas funcionasen correctamente. Las tres pruebas que se realizaron con las nuevas alarmas fueron las siguientes:

- **Vial no llega a la mesa:** Para comprobar que la alarma se activaba correctamente y que aparecía el aviso en el HMI, se probó un ciclo con una de las fotocélulas de control de posición desactivada. Con ello, se podía averiguar el comportamiento del ciclo en caso de no llegar el vial a su posición.
- **Pistón no sale:** Para estudiar si esta alarma ejercía su función con normalidad, se hizo un nuevo experimento desactivando el detector magnético de la posición de salida.
- **Pistón no entra:** Se procedió del mismo modo que para comprobar el funcionamiento de la anterior alarma.

Con estas pruebas se quería verificar que las actuaciones ante fallo fuesen las deseadas, además de comprobar que las alarmas sonoras y lumínicas se activaban correctamente y que en la pantalla del HMI se especificase cuál era el fallo.

### 11.4. Actuación ante fallos de fotocélulas de posición

Con las modificaciones y verificaciones mencionadas en los anteriores apartados, el programa estaba listo para implementarse en producción. Aun así, se quiso añadir una nueva medida de seguridad.

Por más que las fotocélulas de posición de viales estuviesen calibradas para detectar simplemente la primera fila de la mesa, podía ser que con el movimiento de esta tras avanzar una fila, quedasen activadas, dando un falso positivo. De darse este fallo, el programa entraría en contradicción ya que por un lado, consideraría que todavía no habría entrado ningún inyectable a la mesa, pero por el otro, estaría detectando que un vial estaría posicionado. Además, tampoco existiría un control cuando realmente el vial llegase a la fotocélula en cuestión, puesto que previamente ya estaría activada.

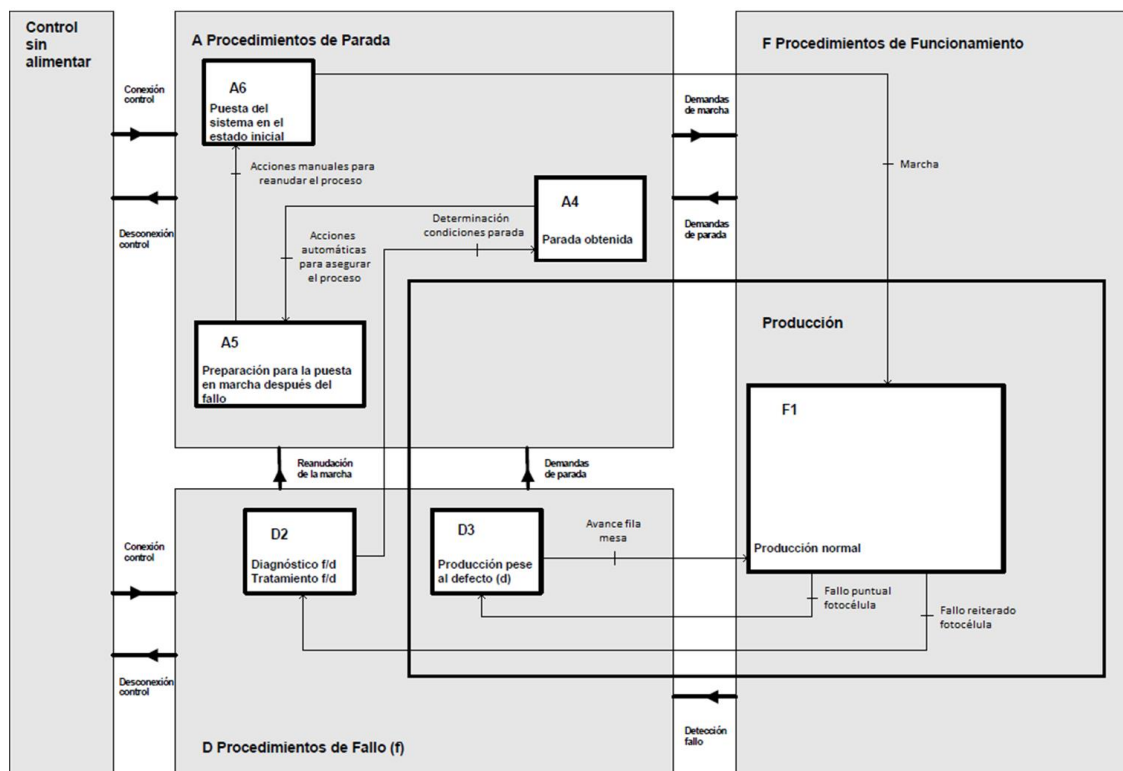
Este apartado también podía incluirse en el módulo de modos de marcha de la Guía GEMMA.

Ya que no se sabía si este hecho sería común, se decidió incorporar una primera solución y además añadir un contador de fallos, con el fin de decidir si la propuesta era correcta o si se debía plantear una nueva alternativa.

La propuesta de actuación frente a este fallo fue que si había una fotocélula de posición activada durante un tiempo mientras que mediante el conteo de entrada de viales a la mesa se consideraba que no habían suficientes viales como para que esto sucediese, se activaría el pistón referente a esa fotocélula para evitar tener que tratar con la fotocélula en fallo durante esa fila. Tras acabar de llenarse la fila, que contendría un número inferior de viales y diferente según la fotocélula que hubiese fallado (de fallar la primera fotocélula solo faltarían dos viales mientras que si fallaba la tercera y todavía no había entrado ningún vial, esa fila únicamente tendría dos tiras), se continuaría con el mismo ciclo. Si este hecho era puntual, era preferible tener una fila con menos de ocho tiras antes que tener que parar la instalación y que un operario comprobase la fotocélula, o arriesgarse a un mal funcionamiento.

Mediante el uso de marcas, se memorizaría que había sucedido un fallo puntual con tal de provocar una alarma si esto volvía a suceder con uno de los dos siguientes avances de fila de mesa, ya que significaría que no era un caso aislado sino que la fotocélula no funcionaba correctamente. En este caso sí que sería necesaria la entrada de un operario de mantenimiento a la instalación para inspeccionar (y arreglar si fuese necesario) la fotocélula.

Para este caso, se combinaría la producción pese a fallo del modo D3 con la actuación frente a fallos especificada en el *Apartado 6.2.3*.



**Figura 11.4:** Diagrama Guía GEMMA para el control de fallos de las fotocélulas

Esta situación de fallo modificaría el GRAFCET G1 ya que añadiría una nueva condición para la salida de los pistones, independiente del resto (error puntual) mientras que si el error fuese reiterado, se comportaría del mismo modo que los otros fallos del *Apartado 6.2.3*, representados por el GRAFCET de conducción G100.



Para la gestión de este tipo de fallos, se creó un nuevo bloque de funciones, el FB21, llamado FB\_FotocelulasFallo, con un bloque de datos de instancia asociado, el DB24, apodado DB\_FotocelulasFallo. La llamada del FB21 se realizaría desde la FC20.

## 12. Coste final del proyecto

Una vez finalizado el proyecto, se pudo calcular de forma estimativa el coste que tendría incluyendo además del presupuesto de la empresa proveedora de material e instalación, el precio de las horas dedicadas y de la parte proporcional de los programas informáticos utilizados.

Para el tiempo de dedicación, con un coste horario relativo al de un ingeniero de prácticas, se ha tenido en cuenta la toma de datos (60 horas), diseño de la mejora (150 horas), investigación y formación (100 horas), montaje (10 horas), puesta en marcha (30 horas) y redacción del informe (100 horas), resultando un total de 400 horas. En cuanto a la supervisión y soporte por parte del programador de la empresa, se contabilizó teniendo en cuenta la formación facilitada (30 horas), la supervisión de los programas (15 horas) y el soporte en la puesta en marcha (10 horas), llegando a un total de 55 horas.

Por otro lado, había que considerar una parte proporcional del coste de los programas informáticos utilizados. En primer lugar, la licencia del entorno de programación de Siemens (STEP7 y TIA Portal) tiene un precio de aproximadamente 2500€. Dado que es un programa que se usa constantemente en la empresa, se aplica un coste de un 5% del total en relación a este proyecto, por lo que el coste atribuido a estos programas sería de 125€. En cuanto al SolidWorks utilizado para la realización de los planos, al ser la licencia para estudiantes de la *Universitat Politècnica de Catalunya* gratuita, el coste era nulo en este aspecto.

**Tabla 12.1: Coste final proyecto**

	Precio	Unidades	Cantidad	Total
Material e instalación	3527	€/máquina	1	3527€
Trabajo ingeniero prácticas	8	€/hora	450	3600€
Trabajo programador experto	30	€/hora	55	1650€
Licencias informáticas	125	€/unidad	1	125€
<b>Total</b>				<b>8902€</b>

## 13. Impacto medioambiental

En cualquier proyecto es imprescindible realizar una Evaluación del Impacto Ambiental. Cabe destacar que, en el presente trabajo, la repercusión sobre el medio ambiente era mínima.

En primer lugar, la instalación no producía emisiones de gases contaminantes. Por otro lado, los cambios efectuados a raíz del proyecto tampoco iban a provocar ningún tipo de contaminación al agua. El punto que sí que podía tener cierta relevancia era el de los residuos.

Como aspecto positivo, la implementación de este proyecto conseguiría reducir el rechazo de tiras de inyectables, ya que se iba a evitar que se dañasen a causa de un aplastamiento con las bandejas, debido a que los viales ya no se colocarían incorrectamente. Esto permitiría minimizar una parte de los residuos que se generan en la planta, dado que, al desechar una tira dañada, se pierde el polietileno que forma las botellas. Asimismo, una optimización del ciclo productivo conlleva indirectamente una reducción del consumo energético, puesto que, al ser más eficiente, precisa de un menor número de tiras producidas para llegar a obtener la misma cantidad de unidades finales.

Por otra parte, la cara negativa era que los elementos utilizados para la automatización (fotocélulas, cilindros neumáticos, detectores magnéticos, etc.) tienen una vida útil, generalmente de entre cinco y diez años, por lo que, a la larga, acaban generando nuevos residuos. Estos elementos electrónicos están compuestos por materiales que requieren un tratamiento especial para su reciclaje, como puede ser el plomo, mercurio o arsénico. Con la rápida evolución de la tecnología, cada vez son más abundantes este tipo de restos, por lo que es importante que se tome consciencia de que hay que hacer todo lo posible por reciclarlos. Para ello, conviene que la empresa tenga unas directrices muy marcadas en cuanto al tratamiento de residuos.

Los puntos limpios son instalaciones municipales que se encargan de gestionar los residuos que no pueden depositarse en los contenedores urbanos de recogida selectiva. Por lo tanto, la empresa debe facilitar este recurso. Este era el caso de la empresa Fresenius Kabi S.A.U.

Por último, en cuanto a la estructura del montaje, al ser elementos de uso común (perfiles de aluminio, pletinas o soportes), en el caso de dejar de ser necesarios en la instalación, sería sencillo encontrarles un nuevo uso. De lo contrario, debido a que son piezas de acero o aluminio, tienen un fácil método de reciclaje. Aunque el aluminio se puede tirar en el contenedor amarillo, ambos pueden ser también depositados en un punto limpio, ya que cuentan con la opción de reciclar metales.

## 14. Resultados obtenidos

Una vez realizadas todas las pruebas para la puesta en marcha, el dispositivo estaba preparado para su incorporación en el ciclo productivo normal. Antes de validarse su cambio, debía pasar un control por parte del departamento de Calidad para que verificase su correcto funcionamiento y su sistema de seguridad.

Para efectuar el cambio, debería esperarse a una parada de la inyectora que permitiese el montaje de las modificaciones y la carga de los programas. El cambio definitivo se planificó para el verano, ya que la planta permanecería diversos días consecutivos sin producir, hecho que permitiría una transición más gradual, por si se requerían depuraciones del programa por fallos no considerados.

Estas fechas eran posteriores a la finalización de este proyecto, por lo que no se pudo contar con valores exactos de la repercusión de esta mejora una vez incorporada en producción. Además, era necesario un periodo largo de toma de datos para comparar con los valores anteriores (mínimo dos semanas, que fue el tiempo dedicado previamente).

Aun así, por las múltiples pruebas realizadas a lo largo de la puesta en marcha, se esperaba que la incorporación corrigiese completamente el problema de la incorrecta colocación de los viales por parte del robot, ya que no había dado ningún problema durante este proceso. Así bien, se contaba con poder recuperar finalmente la inversión, tal y como se había estimado.

Por otro lado, el hecho de trabajar en esta instalación había propiciado la incorporación de otras mejoras: depuración de partes del programa; programación e instalación de un separador de tiras a la salida del envasado (de camino a la mesa) para evitar acumulaciones en la entrada; adición de un control (y rechazo sin parar la instalación) por si llegaba alguna tira de inyectables con menos de cinco botellas, es decir, rota; o incluso la implementación de mecanismos de seguridad como el bloqueo de las cintas de rechazo de viales una vez abierta la puerta de seguridad de la instalación, la incorporación de una alarma al abrirse la protección en el rechazo de viales de la entrada, o el cambio de posición del armario del robot para facilitar el paso de los operarios a la zona trasera de la estación de carga.

Es por ello que los resultados obtenidos al finalizar este proyecto han sido muy positivos de cara a la empresa, justificándose su inversión.

## 15. Propuestas de mejora

Debido a que este proyecto está pensado para permanecer indefinidamente en la planta, se podían plantear mejoras para un posterior proyecto, con la finalidad de perfeccionar el proceso de carga:

- **Ajuste de la posición de colocación de los viales en la bandeja:** Uno de los aspectos interesantes a verificar era el correcto funcionamiento de la comprobación de posición que realizaba el robot una vez colocada una nueva bandeja sobre el bloque que estaba cargando. Al poner la bandeja, gracias a un conjunto de fotocélulas, revisaba que la posición de las cuatro esquinas fuese la correcta, parando la marcha y activando una alarma en caso contrario. Una buena mejora sería incorporar la posibilidad de que el robot corrigiese las coordenadas de posicionamiento de viales en función de los datos obtenidos en dicha comprobación.  
El posicionamiento de las bandejas no debería ser incorrecto ya que contaban con topes en la base del bloque y cada una de ellas se acoplaba exactamente mediante cuatro puntos resaltados que tenían en las esquinas. Aun así, de darse este caso, sería interesante que el propio robot pudiese corregir un fallo como tal y evitar que colocase los viales de forma incorrecta por ir a buscar la posición teórica. Esta mejora habría que realizarla por parte del propio programa.
- **Control visual del posicionamiento de los viales:** En segundo lugar, resultaría útil que existiese un control en el posicionamiento de viales en la bandeja. Para este proyecto, los datos se habían tenido que tomar manualmente, identificando a simple vista cuando el robot colocaba incorrectamente un vial. Con tal de contar con otro elemento de control, sería beneficioso para el automatismo si se realizase un control visual de las bandejas al llenarse para identificar si se daba un caso de mala colocación. Por otro lado, se podría estudiar si sería conveniente programar algún tipo de reacción por parte del automatismo para corregir este suceso, ya sea parando la instalación o realizando acciones automáticas. Cabe destacar que este fallo se espera que no se dé una vez incorporado este proyecto, pero siempre es importante tener un control máximo del proceso.

- **Optimización del proyecto actual:** En cuanto al proyecto de mejora de carga, sería necesario un estudio posterior a su implementación para analizar los problemas que pudieran aparecer. Con ello, se podría realizar una optimización del mismo proceso. Uno de los aspectos importantes sería seguramente un ajuste de los temporizadores de espera con tal de reducir el tiempo de ciclo, ya que probablemente sería uno de los aspectos más críticos.

## Conclusiones

Al término de este proyecto, existía un alto grado de satisfacción tanto por parte del estudiante como de la empresa, ya que los objetivos expuestos al iniciar el proyecto habían podido cumplirse.

En primer lugar, se consiguió identificar un aspecto de la línea de producción de tiras de inyectables de 10 ml que generaba problemas, causando múltiples paradas. A partir de ello, se pudo dar con uno de los focos de estos fallos y desarrollar una solución viable tanto técnica como económicamente. Por más que no se pudiera implementar la mejora en el plazo del proyecto, quedó todo preparado para su inminente instalación, con garantías de su funcionamiento.

A raíz de los resultados obtenidos en el proyecto, se decidió que se procedería a implementar la misma solución para otras estaciones de la planta, ya que además de la referente a la BP8 (la otra extrusora de 10 ml), existían más máquinas para formatos de inyectable distinto, que empleaban el mismo método para la carga.

En cuanto a la experiencia adquirida, los objetivos se cumplieron con creces ya que se pudo obtener una amplia formación en la programación de controladores lógicos programables (PLC), en este caso de la marca Siemens, pero que serviría para cualquier proyecto posterior de automatización, dado que las distintas marcas guardan similitud. Dicha formación fue más allá de iniciarse en el entorno de programación de Siemens, lo más importante fue tomar conciencia de la complejidad que tiene un proyecto de automatización, ya que requiere tener presentes todas las situaciones, que muchas de ellas no se descubren hasta su puesta en marcha. La fase de puesta en marcha es la más compleja ya que requiere un alto grado de experiencia, por lo que resultó muy interesante experimentar con un proyecto de este tipo.

Asimismo, la formación no fue únicamente sobre programación de PLC. Otro aspecto muy valioso fue todo el conocimiento adquirido respecto a los elementos utilizados en la instalación como por ejemplo las fotocélulas, los detectores magnéticos o los cilindros neumáticos junto con las electroválvulas. La mejor manera de comprender su funcionamiento es utilizándolos y ajustándolos in situ.

Por último, el hecho de llevar a cabo un proyecto dentro de una empresa permitió al estudiante dar un gran paso adelante en relación a la futura incorporación al mundo laboral. Tratar con proveedores, seguir plazos y presupuestos y trabajar con compañeros tanto de un mismo departamento como de otros, son situaciones que no se dan en el mundo académico y que hay que experimentar con tal de crecer como ingeniero.

## Agradecimientos

Agradecer por su colaboración y dedicación al tutor del presente trabajo, Vicenç Puig, puesto que en todo momento ha demostrado interés, visitando incluso la empresa al comienzo de las prácticas con la finalidad de conocer en primera persona el entorno y proyecto.

Por otra parte, un especial agradecimiento a Jordi Grané, tutor de las prácticas y responsable del departamento de Mantenimiento, por comprender a la perfección la tarea de un tutor, ofreciendo orientación para sacar su máximo rendimiento y a la vez mostrando desde un principio total confianza y facilitando los recursos necesarios para el desarrollo del proyecto.

Por último, dar las gracias a Fran Muñoz por su recepción a la llegada de la empresa, facilitando la adaptación, además de implicarse plenamente en este trabajo, primero con la formación requerida y a partir de entonces, con todo el seguimiento del proyecto ya fuese para aportar conocimientos, propuestas o ayuda en cualquier aspecto de la puesta en marcha.



## Bibliografía

### GRAF CET:

BOIX ARAGONÈS, ORIOL; SUDRIÀ ANDREU, ANTONI; BERGAS JANÉ, JOAN GABRIEL. *Automatització Industrial amb GRAFCET*. Barcelona: Edicions UPC, 1998.

### Guía GEMMA:

PONSA ASENSIO, PERE; VILANOVA I ARBÓS, RAMON. *Automatización de procesos mediante la guía GEMMA*. Barcelona: Edicions UPC, 2005.

### Controladores lógicos programables (PLC) Siemens:

SIEMENS A&D FEA. *Manual de formación para soluciones generales en automatización Totally Integrated Automation (TIA)*. Módulo A1: Totally Integrated Automation (TIA). Alemania: Siemens AG, 2001.

SIEMENS A&D FEA. *Manual de formación para soluciones generales en automatización Totally Integrated Automation (TIA)*. Módulo A3: Introducción a la programación de PLC con STEP7. Alemania: Siemens AG, 2001.

SIEMENS A&D FEA. *Manual de formación para soluciones generales en automatización Totally Integrated Automation (TIA)*. Anexo 1: Fundamentos de la programación de PLC con SIMATIC S7-300. Alemania: Siemens AG, 2002.

SIEMENS A&D FEA. *Manual de referencia SIMATIC*. Lista de instrucciones (AWL) para S7-300 y S7-400. Alemania: Siemens AG, 2004.

SIEMENS A&D FEA. *Manual de referencia SIMATIC*. Diagrama de funciones (FUP) para S7-300 y S7-400. Alemania: Siemens AG, 2006.

SIEMENS A&D FEA. *Manual de referencia SIMATIC*. Esquema de contactos (KOP) para S7-300 y S7-400. Alemania: Siemens AG, 2006.

SIEMENS A&D FEA. *SIMATIC NET System Manual*. Industrial Ethernet / PROFINET Industrial Ethernet. Alemania: Siemens AG, 2016.

SIEMENS A&D FEA. *SIMATIC NET System Manual*. PROFIBUS Network Manual. Alemania: Siemens AG, 2016.

GÚTIEZ CHAGARTEGUI, IÑIGO. *Programación Siemens* [en línea]. Álava: 2012. Disponible en: <<https://programacionsiemens.com/>>.

GARCÍA, DAVID. *infoPLC* [en línea]. Barcelona: 2001. Disponible en: <<http://www.infopl.net/>>.

AUTRACEN. *Blog Autracen* [en línea]. México. Disponible en: <<http://www.autracen.com/blog/>>.

### Neumática:

SERRANO NICOLÁS, ANTONIO. *Neumática práctica*. Zaragoza: Ediciones Paraninfo, 2010.

SMC. *Soluciones Eficientes para la Automatización Industrial* [en línea]. España. Disponible en: < <https://www.smc.eu/>>.

### Sensores:

FACULTAD DE INGENIERÍA (UNIVERSIDAD DE BUENOS AIRES). *Automatización Industrial*. Sensores y Detectores [en línea]. Buenos Aires: Material docente FIUBA, 2001. Disponible en: <<http://materias.fi.uba.ar/7206/Detectores.PDF>>.

CORONA RAMÍREZ, LEONEL GERMÁN; ABARCA JIMÉNEZ, GRISELDA STEPHANY; MARES CARREÑO, JESÚS. *Sensores y actuadores. Aplicaciones con Arduino®*. México: Grupo Editorial Patria, 2014.

CONTAVAL. *Blog Contaval*. ¿Qué tipos de sensores fotoeléctricos existen? [en línea]. Valencia. Disponible en: <<https://www.contaval.es/category/blog/>>.

IFM. *Automation made in Germany* [en línea]. España. Disponible en: <<https://www.ifm.com/es/es>>.

## Anexos

Todo el programa realizado en el proyecto (especificado en el *Apartado 7.2.2*) queda recogido en los anexos del trabajo, mostrándose por el siguiente orden:

- **FC11:** FC\_AlimentaciónMesa.
- **FC20:** FC\_SalidasViales.
- **FC21:** FC\_ConduccionCicloPistones.
- **FB20:** FB\_PistonCiclo.
- **FB21:** FB\_FotocelulaFallo.
- **DB20:** DB\_PasosCicloViales.
- **DB21:** DB\_Piston01.
- **DB22:** DB\_Piston02.
- **DB23:** DB\_Piston03.
- **DB24:** DB\_FotocelulaFallo.
- **DB29:** DB\_HMI.